VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

ZABEZPEČENÁ KOMUNIKACE PRO ZAŘÍZENÍ TYPU SMARTPHONE

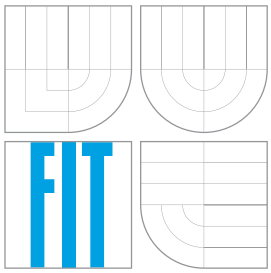
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. DÁVID BOCKO

BRNO 2013

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

ZABEZPEČENÁ KOMUNIKACE PRO ZAŘÍZENÍ TYPU SMARTPHONE

SECURE COMMUNICATION FOR SMARTPHONES

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. DÁVID BOCKO

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. PAVOL KORČEK

BRNO 2013

Abstrakt

Tato diplomová práce se zabývá návrhem způsobu zabezpečení komunikace pro mobilní zařízení typu smartphone. Rozebírá problematiku přenositelnosti aplikací mezi jednotlivými operačními systémy, rozdíly operačních systému a jejich společné znaky. Blíže popisuje architekturu platformy Android a popisuje vrstvy systému. Dále vysvětluje základné způsoby zabezpečení, jak funguje moderní kryptografie a způsob jak je možné ji aplikovat na mobilní platformu. Taktéž popisuje některé moderní technologie, jakými jsou NFC (Near field communication) nebo Bluetooth a přibližuje jejich základní architekturu a způsob jakým zapadají do konceptu aplikace. Tato práce dále popisuje návrh mobilní aplikace a koncept zabezpečení, který je následně implementovaný. V závěru jsou shrnuty dosažené výsledky a případné možnosti rozšíření práce.

Abstract

This master's thesis deals with application design and implementation of secured communication between smartphones. Analyses issues of cross-platform portability between operating systems, its differences and common features. It shows an overview of Android OS architecture and its individual layers. Thesis describes basic principles of secured communication in the modern cryptography and implementation of application for mobile platforms. Also shows an overview of modern mobile technologies such as NFC (Near Field Communication) and Bluetooth. This thesis reveals basic architecture and the way how it fits into concept of secured communication. It also describes the architecture of application and concept of secured communication, which is then implemented. In conclusion is summarization of achieved results and possible extensions of implemented application.

Klíčová slova

Android, zabezpečená komunikace, šifrováni, smartphone, chytrý telefon, Bluetooth, Near field communication, GSM

Keywords

Android, secured communication, encryption, smartphone, Bluetooth, Near field communication, GSM

Citace

Dávid Bocko: Zabezpečená komunikace pro zařízení typu smartphone, diplomová práce, Brno, FIT VUT v Brně, 2013

Zabezpečená komunikace pro zařízení typu smartphone

Prohlášení

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne pod vedením Ing. Pavla Korčeka, ktorý mi poskytol odbornú pomoc a cenné rady. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

.....
Dávid Bocko
22. května 2013

Poděkování

Týmto by som chcel poďakovať všetkým ľuďom, ktorí mi poskytli cenné rady a odbornú pomoc pri vypracovaní tohto projektu. Taktiež by som chcel poďakovať rodičom a priateľke, že vždy stáli pri mne a v ťažkých chvíľach ma dokázali motivovať.

Obsah

Kapitola 1

Úvod

Súčasným trendom vo svete informačných technológii je napredujúci vývoj mobilných systémov a zariadení, ktoré sa stávajú čím ďalej, tým výkonnejšie. Klasické mobilne telefóny sú pomaly vytláčané z trhu a začínajú ich nahradzovať inteligentné telefóny tzv. *smartphones*. Tieto inteligentné telefóny spájajú v sebe mnoho funkcií, ktoré v minulosti museli byť zastúpené samostatnými zariadeniami.

Inteligentné mobilné telefóny dokážu nahradiť fotoaparáty, videokamery a GPS navigácie. Stali sa z nich multifunkčné zariadenia umožňujúce nielen komunikáciu medzi inými mobilnými zariadeniami, ale je možné nimi ovládať aj iné zariadenia, ako sú napríklad zosilňovače domácich kín alebo je možné nimi naštartovať svoje auto [6]. Časom získali rôzne uplatnenia nielen v mobilných technológiách, ale aj ako pomôcky potrebné k bežnému životu. Príkladom je mobilná aplikácia, ktorá umožňuje z telefónu vytvoriť ručnú baterku, s ktorou je možné si posvietiť. Neustály vývoj nových aplikácií zapĺňa medzery na trhu a tým umožňuje vytvoriť prívetivejšie prostredie pre užívateľa.

Na druhej strane môžu aplikácie predstavovať riziko, pretože majú prístup k osobným údajom uloženým v mobilnom telefóne. Neskúsený užívateľ môže pomerne rýchlo prísť o svoje prihlasovacie údaje k bankovému účtu alebo k emailovej schránke. Dáta sú v mnohých prípadoch prenášané v otvorenom tvare a neposkytujú žiadne súkromie užívateľom. Jedným zo spôsobov ako je možné predísť týmto odcudzeniam je použitie zabezpečeného kanálu.

Cieľom tejto práce je vytvoriť a následne implementovať mobilnú aplikáciu, ktorá umožní mobilným telefónom, a teda aj ich majiteľom, bezpečne komunikovať cez rôzne komunikačné kanály. Aplikácia bude taktiež poskytovať aplikačné rozhranie pre iné aplikácie a umožní im bezpečný prenos dát na koncové zariadenie. Cieľom je minimalizovať možné prelomenie šifrovaných dát a znemožniť prípadnému útočníkovi získanie prístupu k dátam. Text práce je vytvorený ako diplomová práca a je pokračovaním semestrálneho projektu.

Práca je rozdelená do niekoľkých kapitol. V prvej kapitole sú popísané operačné systémy, vzájomná prenositeľnosť aplikácií, ich spoločné a rozdielne prvky. Nasledujúca kapitola pojednáva o komunikačných technológiách, ktoré sa používajú na dnešných mobilných platformách a ich bezpečnostné nedostatky. V ďalšej kapitole sú popísané niektoré koncepty a bezpečnostné mechanizmy, ktoré dokážu zabezpečiť komunikáciu, a ktorými je možné vytvoriť zabezpečený kanál. Nasledujúca kapitola obsahuje návrh zabezpečovacej vrstvy medzi mobilnými zariadeniami typu smartphone. Predposledná kapitola pojednáva o niektorých spôsoboch prelomenia, ktorými by sa mohla narušiť implementovaná zabezpečovacia vrstva. V poslednej kapitole je popísané zhodnotenie dosiahnutých výsledkov a prípadné rozšírenia aplikácie.

Kapitola 2

SW podpora pre zariadenia typu smartphone

V tejto kapitole je popísaný prehľad mobilných operačných systémov, spôsoby transformácie aplikácií na inú platformu a stručný prehľad operačných systémov. Bližšie sa zaoberá zvoleným operačným systémom a dôvody, na základe ktorých je vhodnejšie začať vytvárať aplikáciu pod týmto operačným systémom.

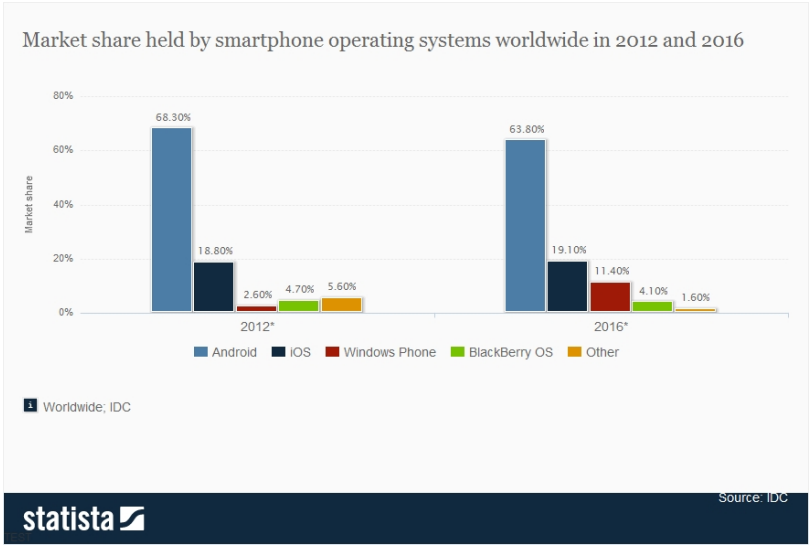
2.1 Prehľad mobilných operačných systémov

Postupným vývojom mobilných telefónov začali vznikať aj operačné systémy, ktoré dovoľujú vytvoriť platformu pre programátorov. Na trh sa dostali rôzny výrobcovia operačných systémov, pričom najväčšiu popularitu získali systémy ako iOS, Android, Windows Phone alebo BlackBerry OS. Tieto operačné systémy úzko súvisia s výrobcami mobilných telefónov. Takmer každý výrobca mobilných telefónov podporuje iba jednu platformu operačného systému a nie je medzi nimi skoro žiadna vzájomná kompatibilita. Užívateľ, ktorý si kupuje mobilný telefón, dostane taktiež so zariadením aj podporovaný operačný systém. Tento operačný systém nie je možné vymeniť za iný operačný systém, ako je tomu u bežných počítačov (OS Linux dokáže bežať na zariadení, na ktorom bežal OS Windows).

Lídrom medzi operačnými systémami je systém Android s najväčším podielom na trhu, za ktorým hneď nasleduje operačným systémom iOS [18]. S veľkým podielom na trhu vznikajú niektoré nedostatky spojené s operačným systémom Android, a tým je veľká diverzita zariadení, ktoré musí tento operačný systém podporovať. U iných výrobcov, ako napríklad u spoločnosti Apple, je operačný systém odladený na konkrétne zariadenie. Návrh aplikácií nemusí brať v úvahu s tým spojené detaily.

2.2 Prenositeľnosť aplikácií

Vytvorená aplikácia pre jeden operačný systém nedokáže bežať pod iným operačným systémom a je nutné ju upraviť podľa požiadavkou danej platformy. Aj napriek tomu existujú určité prostriedky, ktoré umožňujú jednoduchšiu transformáciu na daný operačný systém, prípadne iný programovací jazyk. Jedným z takýchto nástrojov je konvertor zdrojových súborov z jazyka Java do jazyka Objective-C. Tento konvertor nájde uplatnenie hlavne pri transformácii z operačného systému Android na operačný systém iOS [7].



Obrázok 2.1: Prehľad podielu operačných systémov. Prebrané z [18].

Transformácia aplikácie z operačného systému Android na operačný systém BlackBerry OS je už podstatne jednoduchšia ako v prípade iOS. To hlavne z dôvodu, že BlackBerry OS používa rovnaký programovací jazyk (Java) a používa i niektoré spoločné prvky. Na transformáciu aplikácie z operačného systému Android na BlackBerry OS je možné, mimo iné, použiť aj tzv. *Eclipse Plug-in*, ktorý dovoľuje „prebaličkovať“ a prepísať Android aplikáciu na BlackBerry platformu [4].

V súčasnosti taktiež začínajú vznikať poskytovatelia služieb, ktorí zabaľujú vývoj aplikácií pre rôzne operačné systémy do jedného balíčku tzv. *framework*. Vývojár mobilnej aplikácie sa už nemusí zaoberať prenosom na iné platformy, pretože správny beh aplikácie zaistí samotný framework [2]. Jednou z nevýhod vývoja aplikácií v tomto frameworku je vtom, že tieto služby nie sú zadarmo, ako je tomu u vývojových prostredí určených priamo na vývoj aplikácií pre daný operačný systém. Ďalšia z nevýhod vývoja pod frameworkom je v tom, že narastá zložitosť vyvíjanej aplikácie, pretože je nutné ošetrovať rôzne prípady „náhodných“ pádov a zvláštnych správaní aplikácie. [5].

Dalo by sa povedať, že všetky mobilné operačné systémy poskytujú rovnakú funkcionalitu, len je nutné správne použiť pripravené moduly a volania funkcií. Všetky operačné systémy disponujú modulmi na prácu so zariadeniami ako je GPS, Bluetooth, GSM. Taktiež poskytujú mnoho šifrovacích algoritmov a mechanizmov, ktorými sa dá dosiahnuť ciele stanovené v úvode tejto práce.

Z dôvodov popísaných v predošlých odstavcoch, som sa rozhodol zamerať sa na operačný systém Android a celú aplikáciu implementovať pod týmto najčastejšie používaným systémom.

2.3 Architektúra operačného systému Android

Operačný systém Android je plnohodnotný mobilný operačný systém. Vytvorila ho spoločnosť Google a v priebehu času vzniklo niekoľko verzií tohto systému, pričom sa stále zdokonaľuje.

Architektúra systému je rozdelená do vrstiev, ktoré sú medzi sebou vzájomne prepojené.

Každá vrstva využíva služby nižšej vrstvy, nachádzajúcej sa pod touto vrstvou. Základný prehľad vrstiev je nasledujúci [8]:

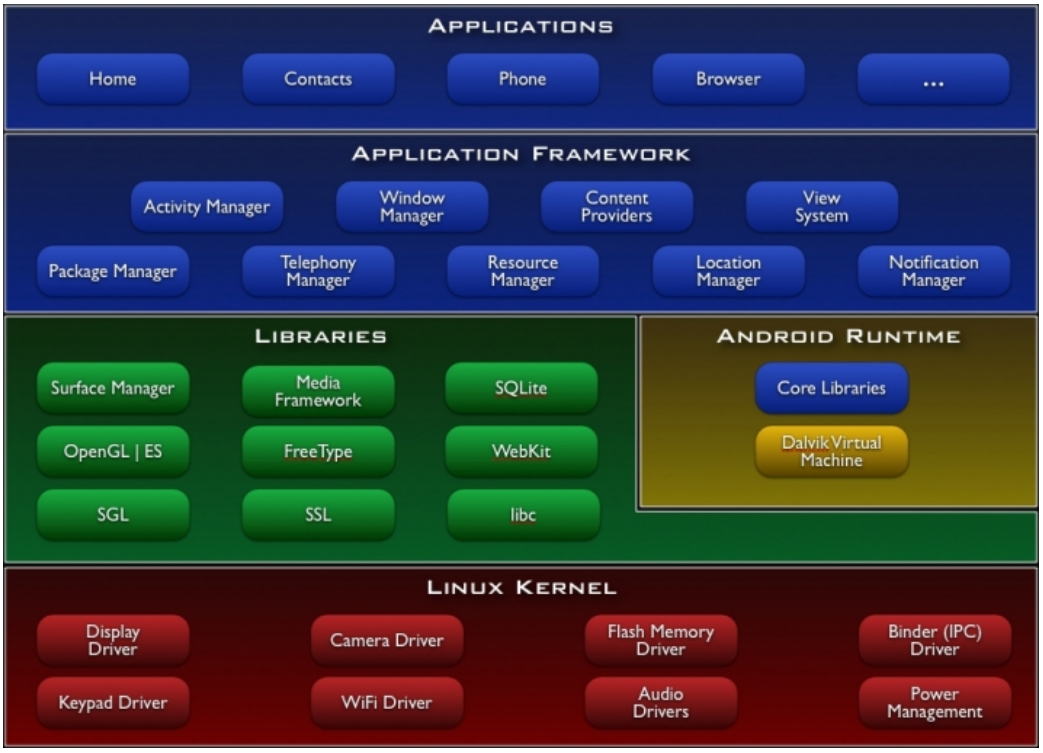
Linux Kernel – Android je postavený na pevnom základe Linuxového jadra. Toto jadro umožňuje Androidu hardvérovú abstrakciu a má na starosti správu pamäťový prostriedkov, správu procesov, správu sietí a iných služieb operačného systému. Užívateľské aplikácie nedovoľujú priamo volať funkcie Linuxu.

Natívne Knižnice – Vrstva nad Linuxovým jadrom, ktorá bola napísaná v jazykoch C alebo C++ a zkompilovaná pre konkrétnu hardvérovú architektúru. Medzi základné knižnice patria *Surface Manager*, *2D* a *3D grafické knižnice*, *Média kodeky*, *SQL databáza*, a *Browser engine*.

Android Runtime – pozostáva z *Dalvik virtual machine* a *Java knižníc*. Dalvik VM je implementácia Javy od Googlu, ktorá je optimalizovaná pre mobilné zariadenia. Všetky vytvorené aplikácie sú spúšťané v Dalvik VM.

Aplication Framework – Vrstva, s ktorou priamo komunikujú aplikácie vytvorené pre Android. Poskytuje vysoko-úrovňové aplikačné rozhranie pre aplikácie.

Aplikačná úroveň – Najvyššia úroveň, kde sa nachádzajú samotné aplikácie. Bežný užívatelia prídu do kontaktu iba z touto vrstvou. Táto úroveň obsahuje niekoľko predinštalovaných aplikácií ako sú telefónny číselník, Zoznam kontaktov, Email, Webový prehliadač a Android Market.



Obrázok 2.2: Architektúra operačného systému Android.

Android zahŕňa niekoľko základných knižníc, ktoré poskytujú väčšinu funkcionality dostupnej zo základných knižníc programovacieho jazyka Java. Každá Android aplikácia beží

vo svojom procese so svojou inštanciou Dalvik virtual machine tak, aby bola čo najefektívnejšia. Dalvik VM spúšťa súbory v tzv. *Dalvik Executable (.dex)*, ktorý je optimalizovaný pre minimálnu pamäťovú stopu. Preklad zdrojových súborov je pomocou Java kompilátoru, ktorý využíva tzv. „dx“ na transformáciu do .dex formátu [8].

2.4 Popis aplikačného frameworku

Ako bolo spomenuté v úvode tejto kapitoly, Android aplikácie sú písané v programovacom jazyku Java a od toho sa odvíja mnoho konceptov, ktoré boli zanesené od tohto jazyka. Bežná Android aplikácia je preložená pomocou Android SDK nástrojov a so všetkými zdrojovými súbormi je vložená do Android balíčku označeného sufixom *.apk*. Spustiteľný kód v jednom *.apk* súbore je považovaný za jednu aplikáciu, ktorú Android zariadenie inštaluje vo svojom prostredí. Android balíček *.apk* obsahuje mimo iné aj súbory *.dex*, ktoré využíva Dalvik VM [8]. Aplikácia, ktorá je úspešne nainštalovaná na zariadení, začína žiť svoj vlastný život vo svojej bezpečnostnej karanténe (application sandbox), kde je izolovaná od ostatných aplikácií [8]:

- Operačný systém Android je multi-uživateľský Linuxový systém, v ktorom je každá aplikácia chápaná ako iný užívateľ.
- Operačný systém priradí vždy každej aplikácii jedinečné Linuxové užívateľské ID číslo, ktoré je pre samotnú aplikáciu neznáme. Všetky súbory asociované s touto aplikáciou majú priradené systémový prístup tejto aplikácie tzn., že nie je možné čítať dáta zo súboru inou aplikáciou.
- Každý proces má svoj vlastný virtuálny stroj, čo znamená, že aplikačný kód beží izolovaný od ostatných aplikácií.
- Štandardne beží každá aplikácia v samostatnom Linux procese. Operačný systém spustí proces keď ktorákoľvek komponenta aplikácie potrebuje byť spustená. V opačnom prípade, keď komponenta už nie je potrebná, je odstránená a nie je udržiavaná ako proces, čím sa uvoľní pamäť pre ostatné aplikácie. Pojem komponenta je bližšie popísaný v nasledujúcej kapitole 2.5.

Operačný systém Android implementuje tzv. *principle of least privilege*, čo znamená, že každá aplikácia má prístup iba ku zdrojom, ktoré aktuálne požaduje ku svojej práci a nezaberá zbytočne ostatné zdroje. Toto vytvára veľmi bezpečné prostredie kde aplikácia nemá prístup ku častiam systému, pre ktoré nedostala prístup.

Aj napriek tomu existujú spôsoby ako môže aplikácia zdieľať dáta s ostatnými aplikáciami a taktiež aplikácia môže zdieľať systémové služby. Jeden spôsob predstavuje mechanizmus, pri ktorom dve aplikácie zdieľajú rovnaké Linux užívateľské ID číslo a sú schopné pristupovať k súborom tej druhej aplikácie. Za predpokladu, že sú aplikácie podpísané rovnakým certifikátom, je možné, aby aplikácie bežali pod rovnakým užívateľským ID číslom a tým zdieľali rovnaký virtuálny stroj. Podpisovanie aplikácii je metóda na zaistenie autorstva aplikácie, kedy sa aplikácia podpíše súkromným kľúčom autora aplikácie [8].

Druhý spôsob zdieľania dát inou aplikáciou je vtom, že si aplikácia sama vyžiada povolenie na prístup k dátovým zdrojom ako sú napríklad užívateľské kontakty, SMS správy, kamera, Bluetooth a iné. Všetky prístupy, ktoré aplikácia potrebuje, musia byť užívateľom schválené počas inštalácie [8].

2.5 Aplikačné komponenty

Aplikačné komponenty predstavujú základné stavebné bloky každej Android aplikácie. Komponenta je bod, cez ktorý môže operačný systém pristupovať do aplikácie. Ale nie každá komponenta musí byť prístupovým bodom do aplikácie. Niektoré komponenty závisia na sebe, ale každá dokáže existovať samostatne a hrá špecifickú rolu v rámci aplikácie, ktorá definuje správanie aplikácie. Existujú štyri základné typy aplikačných komponent a každá slúži inému účelu s iným životným cyklom [8]:

- **Aktivita** – reprezentuje jednu obrazovku s užívateľským rozhraním. Napríklad aplikácia emailovej schránky by sa skladala z nasledujúcich aktivít: aktivita na čítanie emailov, aktivita na zobrazenie všetkých emailov a aktivita na písanie a poslanie emailu. Aktivity pracujú spoločne a formujú jednotný celok aplikácie. Aktivitu je možné spúšťať aj v rámci iných aplikácií, za predpokladu, že je to explicitne povolené v manifest súbore. Manifest súbor je popísaný v nasledujúcej kapitole 2.6.

- **Servis** – je komponenta, ktorá beží na pozadí a vykonáva dlho-trvajúce operácie alebo vykonáva prácu pre vzdialené procesy. Servis komponenta neposkytuje užívateľské rozhranie, ale je možné k nemu pristupovať napríklad z aktivity.

- **Content provider** – spravuje a zdieľa sadu aplikačných dát. Content provider môže byť postavený nad dátami súborového systému, SQLite databáze, na webu alebo na iných perzistentných dátach, ku ktorým má aplikácia prístup. Cez content provider je možné pristupovať k dátam a je možné ich aj meniť, ak to content provider dovoľuje. S content providerom je možné čítať a zapisovať dáta v rámci jednej aplikácie a neumožniť iným aplikáciám prístup.

- **Broadcast receiver** – je komponenta, ktorá reaguje na systémové výzvy. Napríklad systémové výzvy na slabú batériu, vypnutie obrazovky alebo výzvy na zachytený obrázok. Samotná aplikácia môže taktiež iniciovať výzvy napríklad na stiahnutý súbor a poskytnúť ostatným aplikáciám spätnú väzbu. Broadcast receivery nemajú užívateľské rozhranie.

Jednou z vlastností Android aplikácií je, že akákoľvek aplikácia dokáže spustiť inej aplikácii komponentu. Napríklad aplikácia, ktorá požaduje vytvorenie obrázka pomocou fotoaparátu môže spustiť inú aplikáciu, ktorá dokáže zachytiť obrázok a tým nemusí pôvodná aplikácia implementovať proces zachytenia obrázka. Užívateľ má pocit, že aplikácia má proces zachytenia fotografie implementovaný, ale pritom zavolala iba inú komponentu, ktorá dokáže zachytiť a poskytnúť fotografiu [8].

2.6 Manifest file

Pred spustením Android aplikácie musí operačný systém vedieť, že komponenta existuje. Na tento účel slúži súbor *AndoidManifest.xml*. Aplikácia musí deklarovať všetky komponenty v tomto súbore, ktorý musí byť uložený v koreňovom adresári projektu. Okrem deklarácii komponent sa manifest súbor stará o nasledujúce úlohy [8]:

- Identifikácia užívateľských prístupov, ktorá aplikácia požaduje (užívateľské kontakty, prístup na internet, ...).

- Deklarácia minimálneho aplikačného stupňa potrebného pre aplikáciu. Záleží od rozhrania, ktoré aplikácia používa.
- Deklarácia hardvérových a softvérových funkcii požadovaných aplikáciou. (kamera, bluetooth služby, ...)
- Prídavné knižnice, ktoré sú potrebné prilinkovať k projektu.
- A iné...

V prípade, že by aktivita, služba alebo content provider neboli popísané v manifest súbore, nie sú viditeľné operačným systémom a nemôžu bežať. Manifest súbor deklaruje operačnému systému aké prostriedky aplikácia požaduje.

2.6.1 Možnosti šifrovania v OS Android

Operačný systém Android disponuje knižnicou na šifrovanie dát s názvom `javax.crypto`. Táto knižnica poskytuje triedy a rozhranie na šifrovanie užívateľských dát. Implementuje základné protokoly na výmenu kľúčov a šifrovacie a dešifrovanie algoritmy. Knižnica podporuje prúdové, asymetrické, symetrické a blokové šifry.

Operačný systém podporuje mnoho šifrovacích algoritmov a niektoré sú závislé od konkrétneho zariadenia. Medzi základné šifrovacie algoritmy patrí AES a RSA. Pre každý zo šifrovacích algoritmov je možné zvoliť aký veľký bude šifrovací kľúč a tým aj miera zabezpečenia.

Šifrovanie hovorov na operačnom systéme Android je možné iba z časti. Čo sa týka šifrovania hovoru prebiehajúceho cez GSM sieť, tak operačný systém neposkytuje rozhranie na akúkoľvek modifikáciu hlasu počas hovoru. Jediným možným riešením je uskutočnenie telefonického hovoru počas VOIP. Pre tento účel vzniklo niekoľko aplikácií, jednou z nich je napríklad aplikácia RedPhone [14].

Kapitola 3

Popis komunikačných kanálov

V tejto kapitole sú popísané jednotlivé mobilné technológie, ktorými je možné komunikovať pomocou mobilného telefónu typu smartphone. Každá podkapitola bližšie popisuje technológiu a princípy fungovania. Na konci každej podkapitoly sú zhrnuté bezpečnostné riziká, ktorým musí čeliť, prípadne sú navrhnuté možné alternatívy iných technológii. Kapitoly popisujú spôsob prenosu rôznych druhov dát, ako je napríklad prenos hlasu po sieti GSM, prenos krátkych správ SMS alebo prenos dát pomocou technológii Bluetooth alebo NFC.

3.1 GSM

GSM je v súčasnosti najrozšírenejšia mobilná telekomunikačná technológia, ktorú používa skoro každý človek v modernej časti sveta. Pôvod skratky GSM vznikol od *Groupe Spéciale Mobiles*, ktorá bola založená už v roku 1982 a neskôr bol komunikačný systém pomenovaný na *global system for mobile commmnications (GSM)*[17]. Hlavným cieľom GSM bolo poskytnutie mobilného telefónneho systému, ktorý by umožňoval užívateľom sa voľne pohybovať po Európe a poskytoval hlasové služby kompatibilné s ISDN systémom (Integrated Services Digital Network)[17].

GSM je už druhá generácia systému, ktorá nahradila pôvodnú analógovú verziu. V Európe sa začala používať na frekvenciách 890-915 MHz pre uplink a 935-960 MHz pre downlink. Z toho vzniklo označenie *GSM 900*. Okrem GSM 900 taktiež vznikli iné verzie ako je napríklad GSM pracujúce na frekvencii 1800 MHz (1710-1785 MHz uplink, 1805-1880 MHz downlink), ktorý sa volá DCS (digital cellular systems) 1800, a GSM systém pracujúci na frekvencii 1900 MHz (1850-1910 MHz uplink, 1930-1990 MHz downlink), ktorý sa označuje ako PCS (personal communication service) 1900 a je prevažne používaný v Spojených Štátoch Amerických. Okrem týchto základných verzií GSM ešte existujú dve ďalšie verzie a tými sú GSM 400, ktorý sa používa v krajinách s menším počtom osídlenia a verzia GSM-Rail, ktorá je využívaná železničnými spoločnosťami [17].

3.1.1 Mobilné služby

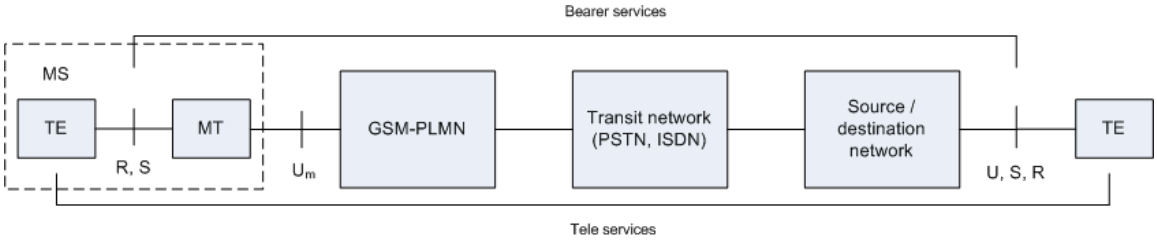
Systém GSM umožňuje integráciu niekoľkých služieb ako sú napríklad prenos zvuku a prenos dát, pričom je pripojená s existujúcimi sieťami. GSM poskytuje tri základné kategórie služieb: *prenosové služby*, *telekomunikačné služby* a *doplnkové služby*. Na obrázku 3.1 sa nachádza zjednodušená schéma GSM siete. Mobilná stanica (MS) je pripojená ku GSM-PLMN (public land mobil network) cez Um rozhranie, ktoré umožňuje pripojenie do GSM siete. Táto sieť je ďalej pripojená ku tranzitnej sieti, ktorú môže tvoriť napríklad ISDN

(integrated services digital network) alebo PSTN (public switched telephone network). K tranzitnej sieti môže byť pripojená prídavná sieť, ktorá sa môže nachádzať pred zdrojovým alebo koncovým terminálom TE [17].

Prenosové služby zahŕňajú všetky služby, ktoré umožňujú transparentný prenos dát medzi rozhraniami jednotlivých sietí. Tieto služby využívajú iba prvé tri vrstvy ISO/OSI modelu [17].

V mobilnej stanici (MS) sa nachádza funkčný blok MT (mobile termination), ktorý sa stará o všetky sieťové úlohy ako sú napríklad TDMA, FDMA, kódovanie atď. a poskytuje rozhranie na prenos dát do terminalu TE, ktorý je nezávislý od siete. Podľa druhu použitého terminálu TE sa odvíjajú aj požiadavky, ktoré sú kladené na rozhranie MT, čím môže byť kladený dôraz na rozšírenie MT [17].

Telekomunikačné služby sú aplikačne závislé, pričom môže vyžadovať všetkých sedem vrstiev ISO/OSI modelu a špecifikuje tzv. *end-to-end* služby.



Obrázok 3.1: Poskytované služby GSM. Prebrané z [17].

Prenosové služby

Ako bolo spomenuté v úvode tejto podkapitoly, prenosové služby využívajú iba tri vrstvy ISO/OSI modelu a stanovujú niekoľko rôznych mechanizmov na prenos dát cez sieť. V pôvodnej verzii GSM, umožňoval iba rýchlosť prenosu do 9600 bit/s a dovoľuje transparentný, netransparentný, synchrónny a asynchrónny prenos dát.

Transparentné prenosové služby využívajú funkcionalitu prvej fyzickej vrstvy ISO/OSI modelu pre prenos dát po sieti. Prenos má konštantnú časovú dĺžku a rýchlosť prenosu, ak nenastane žiadna chyba. Aby sa zabránilo možným chybám v prenose dát, tak je nutné zaviesť určitý druh redundancie dát, čím sa zvýši kvalita prenášaných dát. Implementovaný mechanizmus, ktorý zvyšuje kvalitu prenosu dát sa nazýva *FEC (forward error correction)* a umožňuje zrekonštruovať pôvodné dáta v prípade chyby prenosu. Transparentné prenosové služby nemajú za úlohu nahradiť stratené dáta v prípade tzv. *handoveru* (prerušenia počas prepínania medzi BTS stanicami alebo v prípade rušenia signálu) [17].

Netransparentné prenosové služby používajú protokoly, ktoré využívajú druhú a tretiu vrstvu ISO/OSI modelu, ktoré implementujú opravu chyby a riadenie toku dát. Tieto služby využívajú transparentné prenosové služby s prídavným RLP (radio link protocol) s mechanizmami na znovuzaslanie dát v prípade chyby [17].

Či sa už jedná o transparentné alebo netransparentné služby, GSM špecifikuje niekoľko prenosových služieb, ktoré umožňujú komunikáciu so sieťami PSTN, ISDN a PSPDN (public switched data networking), ako je napríklad X.25, ktorý je dostupný celosvetovo. Prenos dát môže byť plne-duplexný, synchrónny s rýchlosťami 1.2, 2.4, 4.8 a 9.6 kbitov/s alebo plne-duplexný asynchrónny s rýchlosťami od 300 do 9600 bitov/s [17]. Tieto dátové rýchlosti sú

v súčasnosti pomerne pomalé, a preto vznikajú stále nové technológie ako napríklad UMTS sieť, ktorá umožňuje stále väčšie prenosové rýchlosti a väčšiu kvalitu prenosu [17].

Telekomunikačné služby

Systém GSM v prvom rade poskytuje službu na prenos hlasu po sieti, čo zahŕňa šifrovanie hlasu, prenos správ a základnú komunikáciu s terminálmi. Hlavným cieľom GSM bolo zaistenie kvalitného digitálneho prenos hlasu, ktorý by poskytoval minimálne šírku pásma 3.1 kHz telefónneho systému. Pre prenos hlasu sú použite špeciálne kodeky, ktoré transformujú hlasové vzorky [17].

Ďalšou službou, ktorá je poskytovaná systémom GSM je možnosť volať na telefónne číslo v prípade nebezpečenstva. Táto služba musí byť podporovaná všetkými operátormi naprieč Európou a musí byť bezplatná. Táto služba má najvyššiu prioritu a je automaticky prepojená na najbližšie bezpečnostné centrum.

V rámci služieb existujú aj služby založené na textových správach. Asi najznámejšou je tzv. *short message service* (SMS), ktorá umožňuje prenos správy o dĺžke 160 znakov. Táto správa nevyužíva štandardné dátové kanály GSM, ale využíva nevyužité signálové kanály. Príjem a odoslanie správy sa môže diať nezávislé počas prenosu hlasu. SMS správy boli zahrnuté v koncepte GSM už od začiatku, ale nikto ich nepoužíval. Až v polovici 90-tých rokov, kedy sa jej využívanie značne rozšírilo [17].

Nasledovníkom SMS je takzvaná služba EMS (Enhanced message service), ktorá poskytuje väčšiu kapacitu prenášaných znakov (až do 760), formátovaný text alebo prenos animovaného obrázka. Technológia EMS sa ale nikdy moc neujala. Spoločnosť Nokia vytvorila tzv. MMS (Multimedia message service), ktorý nahradil EMS a dokázal prenášať rôzne formáty obrázkov alebo krátke videá.

Doplnkové služby

Tieto služby sú súčasťou GSM, pričom sa môžu odlišovať a byť závislé na konkrétnom poskytovateľovi služieb. Typickými službami sú napríklad identifikácia volajúceho, presmerovanie hovoru alebo tzv. *multi-parti komunikácia*, na ktorej sa môže zúčastniť viacero komunikačných strán [17].

3.1.2 Architektúra GSM

GSM systém je rozdelený do hierarchickej štruktúry, pričom sa skladá z niekoľkých komponent. Zjednodušený nákres je na obrázku 3.2. Nasledujúci popis je prevzatý z [17]:

- *Rádio subsystém* – skladá sa prevažne z rádiových entít, akými sú MS (mobilná stanica) a BSS (base station subsystem)
  - **Base station subsystém (BSS)** – GSM sieť obsahuje mnoho BSS subsystémov, ktoré sú riadené pomocou tzv. base station controler (BSC). BSS sa stará o všetky potrebné funkcie, ktoré zabezpečujú rádiové spojenie s MS a to: kódovanie/dekódovanie hlasu, adaptácia prenosovej rýchlosti do/z bezdrôtovej časti. Okrem BSC, BSS obsahuje niekoľko BTS staníc.
  - **Base transceiver station (BTS)** – BTS stanica obsahuje všetky potrebné rádiové komponenty, ako napríklad antény, zosilňovače signálu, komponenty na spracovanie signálu atď. Tvorí spojovací článok medzi mobilnou stanicou (MS)

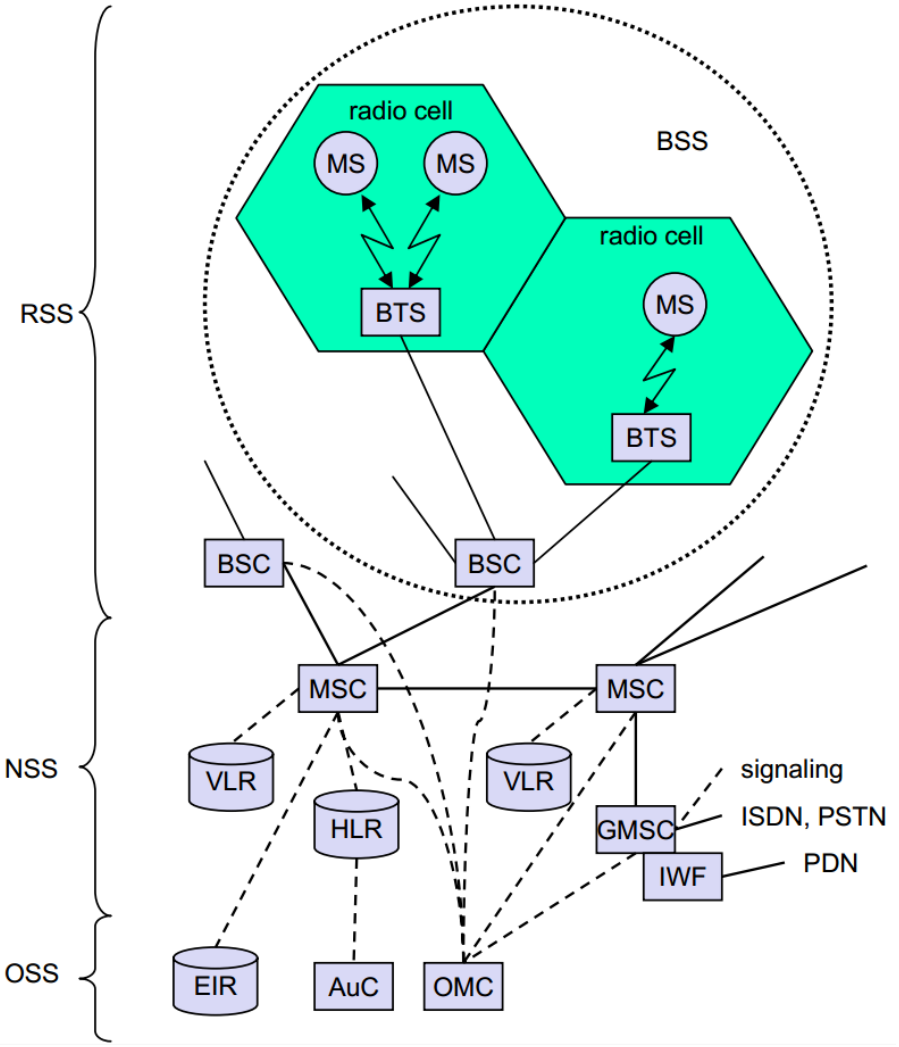
a BSC pomocou rozhraní. Signál dosahuje pokrytie od niekoľko 100 m až po 35 km v závislosti na prostredí (od zastavaných miest až po neosídlené oblasti).

- **Base station controler (BSC)** – spravuje BTS stanice a stará sa o rezerváciu frekvencií, ošetruje handover. BSC taktiež multiplexuje rádio kanály na fixnom sieťovom spojení.
- **Mobilná stanica (MS)** – spája všetky užívateľské komponenty a softvér potrebný pre komunikáciu s GSM sieťou. Aby mohla mobilná stanica komunikovať so sieťou GSM, potrebuje tzv. *subscriber identity module (SIM)*. Tá má v sebe uložený užívateľské dáta, na základe ktorých sa spoplatňujú služby poskytované sieťou. Okrem iného obsahuje aj *personal identity key (PIN)*, *PIN unblocking key (PUK)*, *autentizačný kľúč Ki* a *international mobile subscriber identity (IMSI)*. Mobilná stanica je potom identifikovaná pomocou tzv. *international mobile equipment identity (IMEI)* a využíva SIM k personalizácii mobilnej stanice.

- *Networking and switching subsystem* – predstavuje srdce celého GSM a spája bezdrôtové siete so štandardnými verejnými sieťami. Taktiež sa stará o prepínanie medzi rôznymi BSS, spája funkcie pre celosvetovú lokalizáciu, podporuje spoplatňovanie, účtovanie a roaming medzi rôznymi operátormi. Tento subsystém sa ďalej skladá z nasledovných komponent:
  - **Mobile service switching center (MSC)** - jedná sa o veľmi výkonné digitálne ISDN prepínače, ktoré prepínajú spojenia s inými MSC centrami. Jedna MSC spravuje niekoľko BSC v jednom geografickom regióne. MSC gateway predstavuje zariadenie s prídavným prepojením so sieťami PSTN alebo ISDN. MSC poskytuje funkcionalitu na presmerovanie hovorov, multi-party hovory alebo obrátené spoplatnenie.
  - **Home location register (HLR)** - je jednou z najdôležitejších databáz v GSM. V nej sa nachádzajú všetky užívateľské dáta akými sú mobile subscriber ISDN number (MSISDN), užívateľské služby (presmerovanie hovorov, roaming reštrikcie, GPRS) a international mobile subscriber identity (IMSI). V databáze je taktiež uložená aktuálna poloha zariadenia, aktuálne VLR a MSC.
  - **Visitor location register (VLR)** - je asociované s každým MSC a je to dynamická databáza, v ktorej sú uložené všetky potrebné informácie potrebné pre mobilnú stanicu (IMSI, MSISDN, HLR adresa). V prípade príchodu novej mobilnej stanice sa skopírujú všetky potrebné dáta z HLR do VLR a ďalej sa používa VLR, čím sa predíde častému požadovaniu dát od HLR.

- *Operation subsystem* – obsahuje potrebné informácie pre správnu funkčnosť a údržbu siete. Skladá sa z nasledujúcich komponent:
  - **Operation and maintenance center (OMC)** - monitoruje a riadi ostatné sieťové entity. Medzi základné funkcie OMC patrí monitorovanie priepustnosti siete, hlásenie stavu siete, správa bezpečnosti, účtovateľnosť a fakturácia.
  - **Autentication centre (AuC)** - je autentizačné centrum, ktoré chráni identitu užívateľa a prenos dát. Obsahuje algoritmy na autentizáciu ako aj kľúče pre šifrovanie a generovanie hodnôt pre autentizáciu v HLR. V niektorých prípadoch sú tieto údaje uložené priamo v zabezpečenej časti HLR.

– **Equipment identity register (EIR)** - je databáza na uloženie IMEI čísel, kde sú uložené všetky čísla registrované v danej sieti. Obsahuje taktiež čiernu listinu všetkých ukradnutých alebo zamknutých mobilných zariadení. Na druhej strane obsahuje aj bielu listinu platných IMEI čísel a šedú listinu pokazených zariadení.



Obrázok 3.2: Architektúra GSM. Prebrané z [17].

3.1.3 Bezpečnosť GSM

Systém GSM poskytuje niekoľko služieb, ktoré využívajú tajných informácii uložených v autentizačnom centre (AuC) a v SIM karte. Samotná SIM karta je zabezpečená proti neautorizovanému prístupu pomocou PIN. SIM karta taktiež obsahuje tajný kľúč, ktorý sa využíva na autentizáciu v sieti a šifrovacie procedúry. Bezpečnostné služby poskytované GSM sú nasledujúce:

- Riadenie prístupu a autentizácia - prvým autentizačným krokom je overenie užívateľa voči SIM, kde užívateľ zadá PIN a získa k nej prístup SIM. Druhým krokom je overenie mobilnej stanice voči sieti, ktoré sa deje na základe individuálneho autentizačného

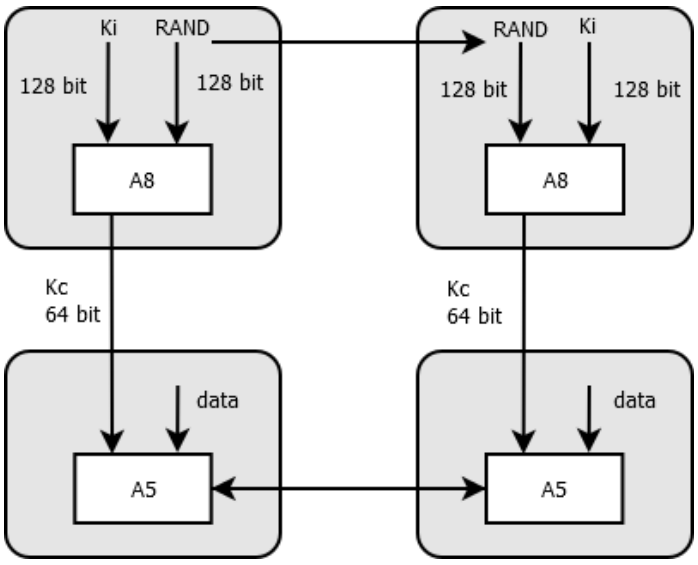
kľúča uloženého v SIM, IMSI a pomocou algoritmu A3. Samotná autentizácia prebieha pomocou výzva-odpoveď metódy, kedy riadenie prístupu vytvorí tzv. náhodné RAND číslo. Toto číslo je zaslané do SIM, kde je pomocou algoritmu A3 a autentizačného kľúča vygenerované nove SRES číslo a to je poslané do MSC a tam sa porovná. Ak čísla odpovedajú, tak je mobilná stanica prihlásená do siete. V opačnom prípade je odmietnutá.

- Dôvernosť - Všetky užívateľské dáta sú šifrované. Po úspešnej autentizácii sú dáta medzi MS a BSS šifrované. To sa týka všetkých dát ako je hlas, dáta a signalizácia. Dôvernosť dát je aplikovaná iba medzi MS a BSS, nie v rámci siete GSM, čím sa neposkytuje šifrovanie dát medzi samotnými mobilnými stanicami.
- Anonymita - je zabezpečená tým, že všetky dáta sú šifrované a nie sú použité žiadne identifikátory, ktoré by odhalili užívateľovu identitu. Namiesto toho GSM používa dočasný identifikátor TMSI, ktorý je generovaný VLR po každej zmene polohy.

V systéme GSM sa používajú tri základné šifrovacie algoritmy. Algoritmus A3 je použitý pri autentizácii, A5 je použitý pri šifrovaní dát a A8 sa používa pri generovaní šifrovacieho kľúča. Šifrovacie algoritmy A3 a A8 si môžu mobilný operátori zvoliť, ale algoritmus A5 je implementovaný v mobilnom zariadení a je identický pre všetkých operátorov.

Šifrovanie dát v GSM

Na zabezpečenie súkromia sa GSM používa šifrovanie medzi MS a BSS. Po úspešnej autentizácii sa dáta šifrujú pomocou algoritmu A5 a kľúča Kc, ktorý sa získa ako výstup algoritmu A8. Do algoritmu A8 vstupuje náhodné číslo, ktoré je vygenerované riadením prístupu a ďalej autentizačný kľúč Ki, ktorý je uložený na SIM a v AuC. Šifrovanie dát bližšie vystihuje nasledujúci obrázok.



Obrázok 3.3: Šifrovanie dát v GSM. Prebrané z [17].

3.1.4 Bezpečnosť UMTS sietí

Nasledovníkom GSM systému je tzv. mobilná sieť tretej generácie (3G), ktorá podporuje širokopásmové služby s rýchlosťou do 2 Mbit/s pre nepohyblivé mobilné zariadenia, 384 kbit/s pri rýchlosti chôdze a 144 kbit/s pri rýchlosti jazdy automobilu. Do tejto kategórie spadá niekoľko rádiových technológii. Jednou z nich je napríklad UMTS (Universal Mobile Telecommunication System), ktorý je súčasťou IMT-2000 (International Mobile Telecommuinications 2000). IMT-2000 predstavuje sadu prijatých doporučení ITU-T, ktorý špecifikuje technológie pre 3G [16].

V roku 1998 vznikol na základe dohody normalizačných organizácii projekt 3GPP (Third Generation Partnership Project), v rámci ktorého sa pripravujú špecifikácie pre UMTS. Väčšina súčasných 3G sietí vychádza zo špecifikácii 3GPP [16].

Technológia UMTS sa podstatne líši od svojho predchodcu. Zmeny nastali hlavne v infraštruktúre siete, pretože všetky prvky siete sú nové. Tým sa zvyšila úroveň zabezpečenia UMTS siete. Zabezpečenie je možné na základe vysokej šírky pásma, čo umožňuje šifrovanie dát bez akéhokoľvek dopadu na rýchlosť prenosu užívateľských dát a na užívateľom vnímanú kvalitu služby [16].

UMTS využíva bezpečnostných prvkov, ktoré boli použité už u GSM sieti, pričom niektoré prvky boli podstatne vylepšené. Najväčším krokom vpred bolo vytvorenie obojsmernej autentizácie voči siete. U GSM využíva iba jednosmernú autentizáciu, čo mohlo viesť k tomu, že útočník mohol využiť falošnú základovú stanicu pre zachytenie identifikácie mobilných užívateľov (IMSI) a vypnúť šifrovanie [16]. Základné bezpečnostné rozdiely medzi GSM a UMTS sú v:

- **Autentizácii** - je založená na modulu USIM (Universal Subscriber Identity Module), ktorý je obdobou SIM u GSM a umožňuje personalizáciu mobilného zariadenia. USIM je prístupný pre užívateľa po zadaní správneho hesla. Pre zaistenie integrity a utajenia sa musí mobilná stanica a sieť dohodnúť na súbore kľúčov. Táto dohoda vzniká vo fáze autentizácie a dohody o kľúčoch (AKA, Authentication and Key Agreement) medzi USIM a HLR. Autentizácia sa deje na základe autentizačného vektoru 3G, ktorý obsahuje parametre akými sú: 128bitové náhodné číslo, očakávaná odpoveď, 128bitový šifrovací kľúč, 128bitový kľúč integrity a token sieťovej autentizácie.

- **Šifrovaní** - šifrovanie a dešifrovanie dát cez rádiové rozhranie sa deje na strane mobilnej stanice aj na strane siete a je vyvolané na základe procedúry AKA. Šifrovanie je funkciou protokolu RLC (Radio Link Control) alebo MAC (Media Acces Control). Využíva sa šifrovanie pomocou algoritmu Kasumi pomocou 128bitového kľúča [20].

- **Integrite dát** - Kľúč pre zachovanie integrity dát sa odvodzuje v priebehu AKA. Samotná integrita dát je poskytovaná iba pri prenose signálnych dát, ale nie v priebehu prenosu užívateľských dát, čo je náchylné na útok pomocou opakovania. Obrana proti útokom opakovaním je možná pomocou aplikačného softvéru.

- **Bezpečnosť aplikácií** - Špecifikácia 3GPP nedefinuje bezpečnostné mechanizmy na aplikačnej úrovni, takže túto úroveň zabezpečenia rieši každý poskytovateľ služieb sám. Jednou z alternatív, medzi ktorými môže poskytovateľ zvoliť, je napríklad IPsec, ktorý zaisťuje transparentnú ochranu cez internet bez závislosti na konkrétnej aplikácii. Šifrovanie v rámci autentizácie a prenosu dát umožňuje ochrániť dáta pred neautorizovaným zásahom a odpočúvaním.

Aj na základe predošlého popisu bezpečnostných mechanizmov v sieti UMTS existujú určité slabiny, ktoré umožňujú nabúrať systém ako celok. Na identifikáciu mobilnej stanice sa používa TSMI, čiže dočasný identifikátor, ktorý sa odvodzuje od IMSI. Ten sa musí poslať v otvorenej forme do VLR/SN pre alokovanie TMSI a žiadny postup autentizácie nie je možný (AKA) bez toho, aby bola známa identita stanice. Takže útočník s falošnou základovou stanicou si môže vyžiadať od mobilnej stanice IMSI v otvorenej forme, ale signalizačný kanál sa šifruje skupinou kľúčov, na základe ktorých môže útočník zúžiť hľadanú identitu mobilnej stanice [16].

3.2 Bluetooth

Bluetooth je veľmi rozšírená komunikačná technológia používaná v mobilných zariadeniach. Svoje využitie nachádza v mobilných telefónoch, tabletoch, notebookoch a iných prenosných zariadeniach.

História Bluetooth sa datuje od roku 1994, kedy švédska spoločnosť Ericsson zahájila štúdie týkajúce sa takzvanej „multi-communicator link“. Projekt bol neskôr premenovaný na Bluetooth. V roku 1998 založili spoločnosti Ericsson, Intel, IBM, Nokia a Toshiba konzorcium Bluetooth s cieľom vyvinúť jednočipovú a lacnú sieťovú technológiu založenú na rádiovej frekvencii. V roku 2001 sa na trh začali dostávať prvé produkty s touto technológiou [17]. V dnešnej dobe sa táto technológia rozšírila nielen medzi mobilné telefóny, ale aj prenosné počítače. Bola vyvinutá na pripojenie zariadení, ktoré sa nachádzajú blízko seba a nahrádza tak káblové pripojenie, a by mal poskytovať podobné zabezpečenie.

Bluetooth je zamerané na prepojenie niekoľkých zariadení. Prepojovací systém je rozdelený do 3 základných kategórii [17]:

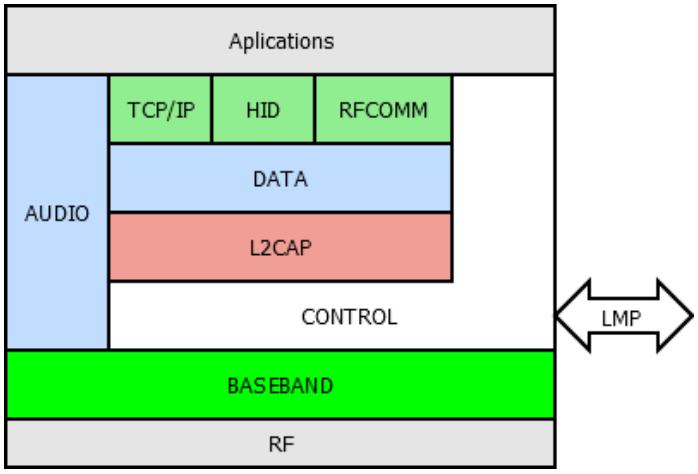
- **Voice/Data Access Point** - Táto kategória predstavuje základné prepojenie Bluetooth zariadení. Jednotlivé zriadenia sa správajú ako prístupové body. Príkladom môže byť mobilný telefón poskytujúci internetové pripojenie a prenosný počítač. Prenosný počítač využíva prístup na internet cez mobilné zariadenie, ktorý môže byť uložený v kapse alebo kabelke.

- **Peripherial Interconnects** - V tejto kategórii Bluetooth slúži ako prepojovací most medzi zariadeniami. Príkladom môže byť prepojenie klávesnice, myší alebo slúchadiel k prenosnému počítaču. Cez tento prepojovací most môžu zariadenia komunikovať aj s inými Bluetooth zariadeniami.

- **Personal Area Networking** - V poslednej kategórii slúži Bluetooth ako ad-hoc sieť. Umožňuje prepojiť zariadenia a bezpečne preniesť medzi nimi dáta.

Architektúra Bluetooth je rozdelená podľa dvoch špecifikácii. Prvou je základná (core) špecifikácia, ktorá popisuje ako funguje samotná technológia. Druhá špecifikácia je zameraná na prepojenie jednotlivých zariadení, pričom využíva prvú špecifikáciu.

Medzi najdôležitejšie časti patrí L2CAP (Logical Link Control and Adaptation Protocol), ktorý zaisťuje niekoľkonásobné logické prepojenia medzi dvoma zariadeniami. Ostatné časti poskytujú možnosti nastavenia pripojenia ako napríklad LMP(Link Management Protocol) a Control vrstva. Popis Bluetooth technológie je čerpaný z [11].

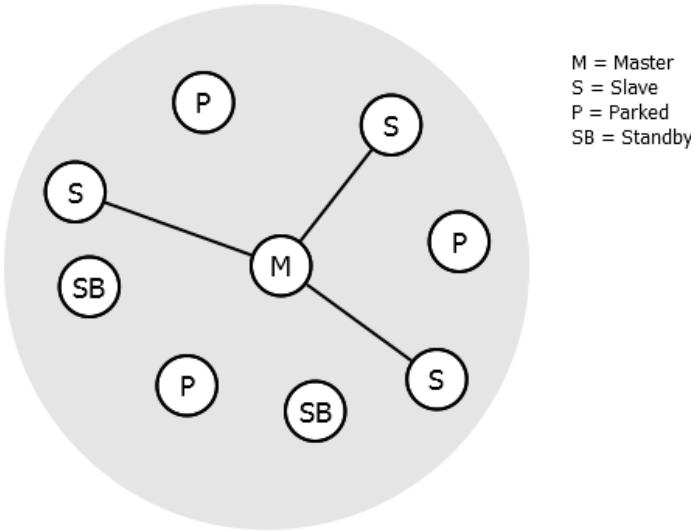


Obrázok 3.4: Architektúra Bluetooth. Prebrané z [11].

3.2.1 Architektúra Bluetooth

Veľmi dôležitým pojmom v oblasti Bluetooth je piconet, ktorý predstavuje kolekciu Bluetooth zariadení, ktoré sú synchronizované na rovnakú skokovú sekvenciu (hopping sequence). V sieti vystupujú zariadenia s rôznou rolou, ktorú hrajú v synchronizácii. Asi najdôležitejšia rola piconetu je master, na ktorý sú napojené ostatné piconety, tzv. slave.

Úloha mastra v piconete spočíva v určení sekvencie striedania nosných frekvencii, na ktorú sa musia slave zariadenia synchronizovať. Každý piconet ma svoj jedinečný tzv. *skokový vzor*. Pokiaľ sa chce zariadenie pripojiť do piconetu, tak sa musí synchronizovať na tento vzor.



Obrázok 3.5: Bluetooth piconet. Prebrané z [17].

Okrem master a slave zariadení sa v piconete môžu nachádzať aj ďalšie dva druhý zariadení. Prvým je tzv. *zaparkované zariadenie (Parked)*, ktoré sa neúčastní v komunikácii napríklad z dôvodu, že nemá pripojenie ale je známe v piconete. Toto zariadenie môže byť znova aktivované v priebehu milisekúnd. Druhý typ zariadenia sa nachádza v pohotovost-

nom režime (Standby), ktoré sa neučastní na komunikácii [17].

Každý piconet obsahuje práve jedno master zariadenie a môže obsahovať až 7 slave zariadení, ku ktorým dokáže pristupovať master zariadenie naraz. Dôvodom pre 7 zariadení je použitie 3 bitovej adresy. V zaparkovanom móde sa môže nachádzať až 200 zariadení. Ak chce zaparkované zariadenie komunikovať a už je obsadených všetkých 7 adries, tak sa jedno slave zariadenie musí prepnúť do režimu zaparkované.

Master zariadenie určuje v piconete sekvenciu striedania nosnej frekvenicie a slave zariadenia sa musia synchronizovať na tento skokový vzor.

Bezpečnosť Bluetooth

Bluetooth poskytuje mechanizmus na autentizáciu a sifrovanie na MAC vrstve, ktorá musí byť rovnako implementovaná na každom zariadení. Jednou zo základných funkcií je použitie výzva-odpoveď rutín pre autentizáciu, prúdová šifra a generovanie relačného kľúča. Spojenie medzi zariadeniami môže byť stanovené ako jednosmerné alebo dvojsmerné, ktoré využívajú autentizáciu. Zariadenia majú možnosť komunikovať aj bez autentizácie. Bezpečnostné prvky v Bluetooth poskytujú iba mechanizmy na ustanovenie spojenia medzi zariadeniami. Väčšia bezpečnosť je poskytovaná až na vyšších vrstvách [16].

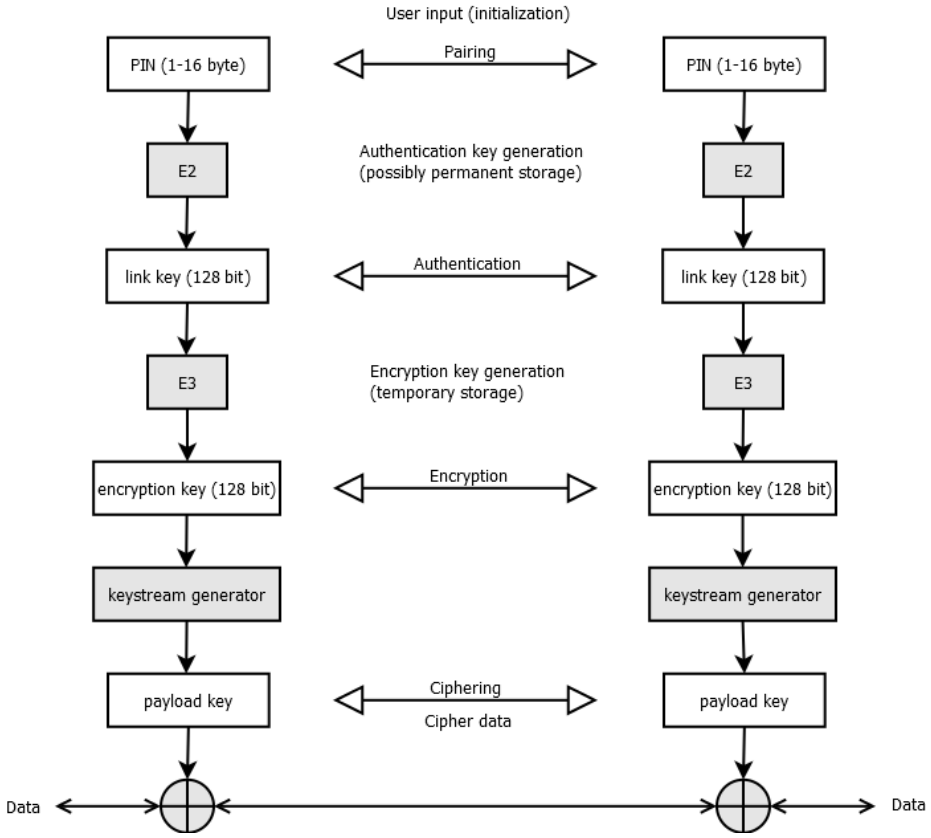
Bezpečnostný algoritmus používa ako vstupné parametre verejnú identitu zariadenia, súkromný užívateľský kľúč a generovaný náhodný kľúč. Pre každú transakciu je vygenerované náhodné číslo v Bluetooth čipe a správa kľúčov je ponechaná na vyšších vrstvách. Jednou zo základných krokov je takzvané spárovanie zariadení, ktoré sa deje v prípade, že dve Bluetooth zariadenia spolu ešte nekomunikovali. Pre spárovanie zariadení je nutné zvoliť PIN, ktorý stanoví dôveru medzi zariadeniami.

3.3 Near Field Communication (NFC)

Near Field Communication je technológia na komunikáciu medzi mobilnými zariadeniami na krátku vzdialenosť (do 20 cm). Jedná sa o vysokofrekvenčnú, bezdrôtovú komunikačnú technológiu. Rozširuje štandard Radio Frequency Identification (RFID) kombináciou rozhrania smartcard a čítačky RFID čipov do jedného zariadenia [15]. Tento štandard je primárne určený pre mobilné zariadenia a účelom je poskytnúť technológiu na platenie v obchodoch pomocou mobilných zariadení. Štandard NFC je možné použiť v troch špecifických módoch:

1. **Mód emulácie karty** - V tomto móde sa zariadenie správa ako bezkontaktná čipová karta. Zariadenie, ktoré číta dáta zo zariadenia, ktoré emuluje kartu, neodkáže rozlíšiť medzi zariadením a skutočnou čipovou kartou. Tento mód sa využíva najmä pri platbách, kde sa využíva už existujúcej infraštruktúry systému platobných kariet. Jedno NFC zariadenie dokáže v sebe uchovať niekoľko virtuálnych čipových kariet.

2. **Mód čítania (zápisu)** - V tomto móde dokáže NFC zariadenie čítať dáta z RFID alebo smart karty. V závislosti na dátach uložených v karte dokáže NFC zariadenie vykonať určité akcie, ako napríklad po prečítaní karty, na ktorej je uložený URI (Uniform Resource Identifier) adresa, dokáže zariadenie otvoriť internetový prehliadač a pripojiť sa na danú URI adresu. V tomto prípade je NFC zariadenie iniciátor komunikácie s kartou, ktorá nie je napájaná zdrojom napätia. V tomto móde je možný prenos dát rýchlosťou 106 Kbit/s.



Obrázok 3.6: Bluetooth zabezpečenie a protokoly. Prebrané z [16].

3. **P2P mód** - V tomto móde dokážu dve zariadenia nastoliť obojsmerné spojenie na výmenu dát akými sú napríklad výmena kontaktov, párovanie Bluetooth zriadení atď. Pri tomto móde dokáže NFC zariadenie prenášať dáta až rýchlosťou 425 Kbit/s. Stanovenie spojenia medzi dvoma NFC zariadeniami je iniciované zo strany klientského zariadenia, ktoré vyhľadáva hosťovské zariadenie. Následne sa dáta posielajú pomocou NFC Data Exchange Format.

3.3.1 NFC architektúra v mobilných telefónoch

NFC technológia integrovaná do mobilných zariadení pozostáva z dvoch základných integrovaných obvodov. NFC radič sa využíva k analógovo digitálnej konverzii signálu cez blízke spojenie. Takzvaný Host Controler Interface (HCI) dovoľuje hostiteľskému radiču nastaviť operačný mód pre NFC radič a spracovať prijaté alebo odoslané dáta [9]. Predstavuje spojovací most medzi hostiteľským radičom a NFC radičom.

Druhý integrovaný obvod slúži na bezpečnostné účely, a ktorý tvorí bezpečnostný prvok (Secured Element) NFC technológie pri móde emulácie karty. Zabezpečenie je zaistené bezpečnostnými mechanizmami v kombinácii s hardvérom, softvérom, rozhraním a protokolmi. Tento modul pozostáva z dvoch prvkov a to: Java Smart Card a Mifare s úložnou kapacitou 4 Kbyty. Bezpečnostný prvok je spojený s NFC radičom pomocou Single-Wire Protocol (SWP) a spojenie medzi hostiteľským radičom a bezpečnostným elementom zaisťuje ISO 7816 rozhranie [9].

Aplikácie, ktoré využívajú NFC komunikáciu, vo väčšine prípadov obsahujú dôverné

informácie, ktoré je nutné nejakým spôsobom zabezpečiť, aby sa za žiadnu cenu nedostali k prípadnému útočníkovi. Na druhej strane ale vniká problém s uložením týchto dôverných informácii, keďže nesmú byť uložené v otvorenom tvare, ale musia byť nejakým spôsobom zabezpečené. Riešením tohto problému je uloženie dôverných dát v pamäti bezpečnostného prvku. V súčasnosti existuje niekoľko vyhotovení takéhoto prvku, a to [9]:

- **Vstavaný hardvér** - táto alternatíva bezpečnostného prvku sa nedá z mobilného zariadenia odstrániť a tým sa stáva bezpečnejším ako napríklad smart card. Táto časť hardvéru je asociovaná s koncovým užívateľom a nemôže byť presunutá do iného mobilného zariadenia.

- **Secured Memory Card (SMC)** - jedná sa o pamäťovú kartu, ktorá sa dá z mobilného zariadenia odstrániť. Pozostáva z vstavanej pamäťovej karty, vstavanej smart card elementu, a smart card radiča a poskytuje rovnakú úroveň zabezpečenia ako technológia smart card. Poskytuje mnoho výhod oproti vstavanému hardvéru popísanému vyššie ako je napríklad výmeny v prípade potreby alebo obnovenie dát v prípade nového zariadenia.

- **Universal Integrated Circuit Cards (UICC)** - ktorá je taktiež známa pod označením SIM. Je kompatibilná so všetkými smart card štandardmi a zaisťuje integritu a zabezpečenie rôzneho druhu osobných dát.

3.3.2 NDEF

Pre zaistenie správneho prenosu dát medzi NFC zariadeniami alebo NFC tagmi, bolo potrebné stanoviť všeobecný formát dát. Tento formát ma názov *NFC Data Exchange Format (NDEF)*.

NDEF je kompaktný a jednoduchý binárny formát, ktorý dokáže prenášať adresy URL, prenášať osobné údaje formátu vCard a NFC špecifických dát. Taktiež umožňuje NFC funkcionalite jednoducho prenášať dáta každého podporovaného NFC tagu, pretože NDEF schováva všetky špecifické detaily o tagu pred aplikáciou [13].

NDEF správa pozostáva zo sekvencie záznamov. Každý záznam obsahuje informácie podľa typu prenášaných dát napríklad URL, MIME alebo NFC-špecifické dáta, ktorých obsah musí byť špecifikovaný v *NFC Record Type Definition (RTD)* súbore. Typ prenášaných dát a ich veľkosť musí byť špecifikovaný v hlavičke NDEF. Hlavička zahŕňa sekciu typu, dĺžku obsahu dát a môže obsahovať taktiež sekciu pre identifikáciu obsahu, ktorá umožňuje aplikácii identifikovať obsah prenášaný v rámci NDEF záznamu. NDEF správa môže pozostávať z jednej alebo viacero NDEF záznamov. Formát správy je nasledujúci [21]:

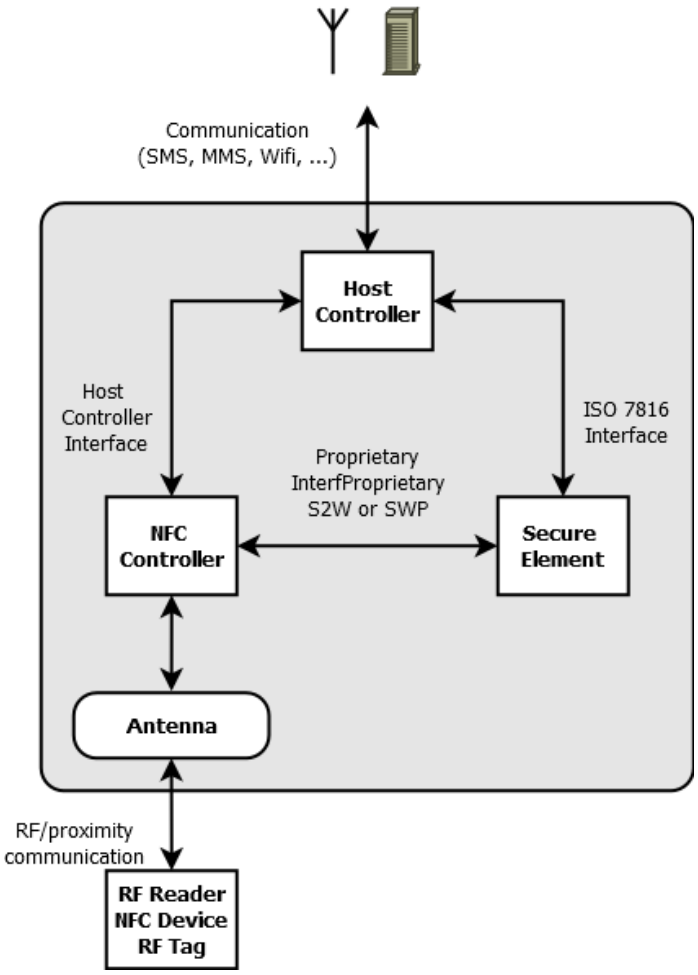
Príznaky majú nasledujúci význam:

- **MB (Message Begin)** - príznak indikujúci začiatok NDEF správy

- **ME (Message End)** - príznak indikujúci koniec správy. V prípade, že je správa fragmentovaná tak tento príznak je nastavený iba v poslednom fragmente správy.

- **CF (Chunk Flag)** - jednobitový príznak indikujúci užitočné dáta. Nachádza sa iba u prvom a prostredných správach.

- **SR (Short record)** - jednobitový príznak, ktorý indikuje skrátenú správu ak je nastavený na 1.



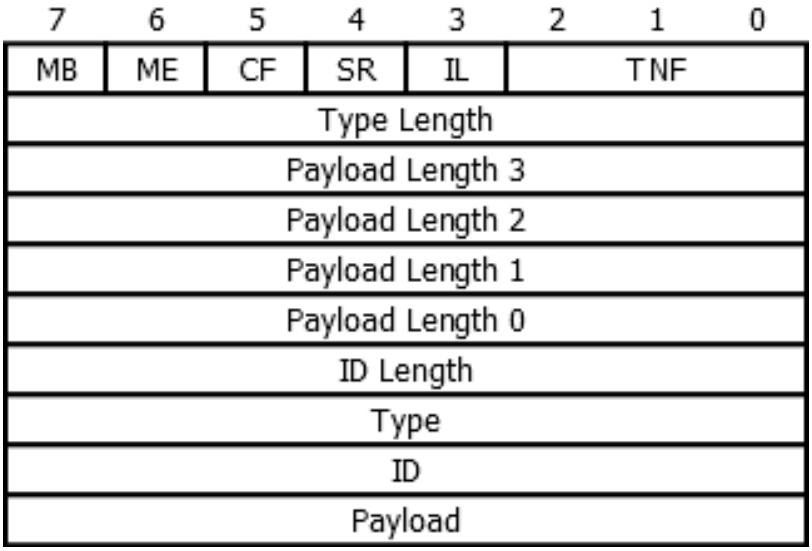
Obrázok 3.7: Architektúra NFC zariadenia. Prebrané z [9].

3.3.3 Bezpečnostné hrozby NFC

Keďže NFC je bezdrôtová technológia vznikajú s tým aj niektoré bezpečnostné hrozby, ako sú napríklad *odpočúvanie dát*, *porušenie integrity dát*, *modifikácia dát*, *ukradnutie prístroja* alebo tzv. *Man-in-the-Middle útok* [12].

Odpočúvanie je metóda, pri ktorej je prenos dát odpočúvaný pomocou tretieho účastníka komunikácie. Dve NFC zariadenia komunikujú pomocou rádiových vĺn a útočník dokáže pomocou antény zachytiť prenášaný signál. Útočník nemusí zachytiť celý dátový tok na to, aby získal súkromnú informáciu. Na zabránenie odpočúvania existujú dve metódy použité v NFC. Prvou je dosah signálu zariadenia NFC. Na to, aby zariadenie dokázalo zachytiť signál dvoch komunikujúcich zariadení, musí byť tretie zariadenie dostatočne blízko. Druhý spôsob je, že NFC zariadenia komunikujú pomocou zabezpečeného kanála. Keď je nastolené spojenie medzi dvoma zariadeniami, tak je komunikácia šifrovaná a iba autorizované zriadenie ho dokáže dešifrovať.

Porušenie integrity a modifikácia dát vzniká, keď chce útočník zmanipulovať dáta, ktoré sú posielané pomocou NFC alebo úmyselne poruší integritu prenášaných dát a tým nedokáže prijímané zriadenie správne prečítať dáta. Pred porušením integrity sa niektoré NFC zariadenia bránia tým, že „počúvajú“ komunikáciu a snažia sa identifikovať útoky spojené



Obrázok 3.8: Formát NDEF správy. Prebrané z [21].

s integritou dát. Po identifikácií útoku zabránia prenosu.

Žiadne šifrovanie nedokáže ochrániť užívateľa pred ukradnutím zariadenia. Teoreticky by zlodej mohol použiť NFC zariadenie pri platení v obchode. Aby sa niečomu takému predišlo, existujú niektoré jednoduché prostriedky ako je napríklad uzamykanie obrazovky telefónu pomocou hesla. Tým je možné zabrániť k prístupu k citlivým informáciám.

Man-in-the-Middle útok využíva podobného princípu ako v predošlom prípade. Medzi dvoma komunikujúcimi zariadeniami je ešte jedno zariadenie, ktoré preposiela prijaté dáta vždy tomu druhému zariadeniu. Tento druh útoku sa vyskytuje vzácne a je ťažšie uskutočniteľný. Aby sa tomu predišlo, tak zariadenia by mali byť v tzv. *active-passive párovaní*. To znamená, že zariadenie dostane informácie a druhé zariadenie ho odošle namiesto toho, že dve zariadenia dostanú preposlanú informáciu.

Kapitola 4

Princíp zabezpečenej komunikácie

Nástupom mobilných operačných systémov vznikajú aj s nimi spojené bezpečnostné riziká, ktorým musia čeliť. Ešte v dnešnej dobe existuje mnoho sietí, ktoré nie sú vôbec alebo len nedostatočne zabezpečené. Užívatelia by si mali dávať pozor na svoje súkromie a takýmto sieťam sa vyhýbať. Kľúčovú úlohu hrá aj správne nastavenie mobilného telefónu ako napríklad nastavenie uzamykania obrazovky alebo nastavenie, ktoré zakáže automatické pripájanie na internetovú sieť.

Postupným vývojom stále nových aplikácií vznikajú aj stále nové hrozby, ktoré môžu ohroziť bežného užívateľa. Jednou takou hrozbou je aj nástup škodlivého softvéru na mobilné platformy. Podľa článku [3], nastupujú dva druhy škodlivého softvéru na Android market. Jedným z nich sú tzv. *trojské kone*, ktoré napríklad posielajú SMS správy na platené služby. Druhým typom škodlivého softvéru je tzv. *spyware*, ktorý zbiera informácie od užívateľa (kontaktné informácie, hesla atď.) a posiela ich tvorcovi spyware-u. Podľa [3] existovalo okolo 14 923 škodlivých softvérov na mobilných platformách detekovaných medzi Aprílom a Júnom roku 2012. Z čoho vyplýva, že je veľmi vhodné používať aj antivírové programy, ktoré dokážu identifikovať škodlivý softvér a prípadne ho odstrániť.

V tejto kapitole sú popísané základné princípy zabezpečenej komunikácie a spôsoby akými je možné dosiahnuť bezpečnej komunikácie. Aby sa mohlo prehlásiť, že komunikácia je zabezpečená, je nutné zaistiť základné stavebné požiadavky akými sú: *autentizácia, autorizácia, dôvernosť, integrita a nepopierateľnosť*. Toho je možné dosiahnuť pomocou modernej kryptografie.

Moderná kryptografia je rozdelená na tri základné celky. Jeden celok predstavuje šifrovanie pomocou *symetricej kryptografie*, druhy celok predstavuje *asymetrická kryptografia* a tretí celok predstavuje *hašovacia funkcia*.

4.1 Symetrická kryptografia

Symetrická kryptografia používa na zašifrovanie a dešifrovanie jeden spoločný kľúč, ktorý musia všetky komunikujúce strany udržať v tajnosti aby nedošlo k jeho prezradeniu. Predstaviteľmi šifrovacích a dešifrovacích algoritmov sú napríklad AES, DES, 3DES alebo IDEA. Na vysvetlenie základných pojmov týkajúcich sa symetrickej ktyptogafie sa v ďalšom texte budem venovať algoritmu *Advanced Ecryption Standard (AES)*.

Šifrovaci algoritmus AES šifruje bloky dát o veľkosti 126 bitov a používa kľúče o rôznych veľkostiach (128, 192, 256 bitov). Algoritmus nemá slabé kľúče, ktoré by podlomili jeho bezpečnosť a nie je žiadná reštrikcia pri výbere zabezpečovacieho kľúča [1]. Blokové

šifrovacie algoritmy akými sú AES (alebo DES) môžu pracovať v rôznych režimoch: ECB (Electronic Code Book) alebo CBC (Cipher Block Chaining). Režim ECB má jednu veľkú nevýhodu, pretože rovnaký blok textu vedie k rovnakému bloku šifrovaného textu. CBC ešte používa inicializačný vektor, ktorý predstavuje postupnosť náhodných bitov a používa sa ako vstup do algoritmu spolu s textom. Inicializačný vektor nemusí byť tajný ale nemal by byť predvídateľný [16]. Pseudo kód algoritmu je nasledujúci [1]:

```
Cipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
  begin
    byte state[4,Nb]
    state = in
    AddRoundKey(state, w[0, Nb-1])
    for round = 1 step 1 to Nr-1
      SubBytes(state)
      ShiftRows(state)
      MixColumns(state)
      AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
    end for
    SubBytes(state)
    ShiftRows(state)
    AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])
    out = state
  end
```

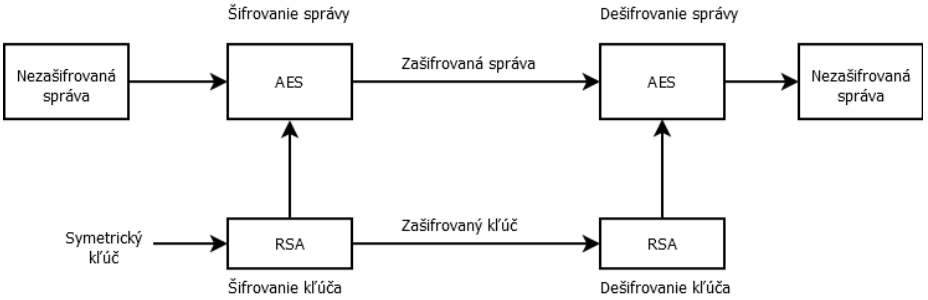
Jednotlivé skratky majú nasledujúci význam:

- **State** - Výsledok stredne ťažkej šifry, ktorý obsahuje 4 riadky a **Nb** stĺpcov.
- **Nr** - Počet cyklov šifry. Pre šifru AES to je Nr=10, 12 alebo 14.
- **Nb** - Počet stĺpcov, ktorý berie v úvahu stav šifry.

Základný algoritmus je pomerne jednoduchý a skladá sa z týchto častí [1]:

- **Pridanie podkľúča** - Každý byte stavu je kombinovaný s podkľúčom za pomoci operácie XOR nad všetkými bitmi.
- **Substitúcia bytov** - nelinárny substitučný krok, kde každý byte je nahradený iným bytom podľa vyhladávajúcej tabuľky. Využíva sa pri tom tzv. *S-box* tabuľka.
- **Posunutie riadkov**- krok, v ktorom je každý riadok postupne presunutý o určitý počet krokov. Prvý riadok sa neposúva.
- **Pomiešanie stĺpcov** - Transformácia, pri ktorej je postupne zkombinujú postupne štyri byty v každom stĺpci.

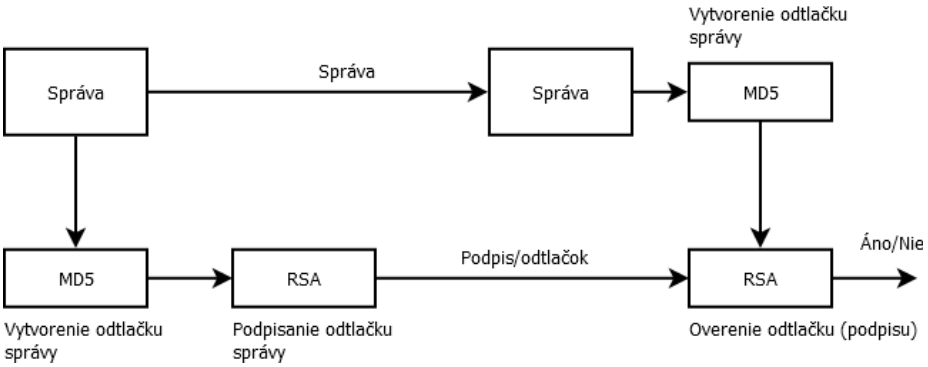
Najväčšia sila symetrickej kryptografie spočíva v kombinácii s asymetrickou kryptografiou. Pre zašifrovanie správy sa používajú rýchle symetrické algoritmy s tým, že sa symetricky kľúč zašifruje asymetrickou šifrou pomocou verejného kľúča príjemcu. Zašifrovaný kľúč so zašifrovanou správou sa spoločne pošle príjemcovi. Tento mechanizmus sa používa pri elektronickom podpise.



Obrázok 4.1: Zabezpečenie dôvernosti správy.

4.2 Asymetrická kryptografia

Základ tvoria dva zabezpečovacie kľúče a to verejný a súkromný kľúč. Verejný kľúč je dostupný každému užívateľovi a súkromný kľúč musí ostať v tajnosti. Verejný a súkromný kľúč tvoria jedinečný vzájomne korešpondujúci pár, pričom dáta sa šifrujú jedným a dešifrujú druhým. Asymetrické šifrovanie slúži k ochrane prenášaných dát, ale nie k autentizácii pôvodu správy, pokiaľ bol použitý dostupný verejný kľúč [16]. V prípade, že dôjde k prezradeniu súkromného kľúča, tak je nutné aby sa o tom dozvedeli komunikujúce strany a vygenerovali si nové kľúče.



Obrázok 4.2: Elektronický podpis.

V súčasnosti existuje niekoľko asymetrických šifrovacích algoritmov. Medzi najznámejšie patrí RSA, Diffie-Hellman, Knapsack, DSS (DSA) alebo ECDSA. Veľkou výhodou asymetrického šifrovanie je relatívne jednoduchá správa šifrovacích kľúčov, pretože pre distribúciu verejných kľúčov nie je treba zabezpečenej komunikácie.

Jedinou veľkou nevýhodou asymetrických algoritmov je, že sú pomerne pomalé a nie sú vhodné na šifrovanie veľkých objemov dát. Tento problém je možné vyriešiť spoločne so symetrickou kryptografiou, ktorá je podstatne rýchlejšia.

V predošlom texte boli spomenuté mechanizmy, ktoré pokrývajú iba časť bezpečnostných funcií. Na zaistenie integrity je nutná tzv. *jednosmerná funkcia*, ktorá sa niekedy volá *hashovacia funkcia*.

4.3 Jednosmerná hashovacia funkcia

Jednosmerná hashovacia funkcia zaisťuje tzv. *odtlačok správy*, ktorý ochraňuje integritu správy pomocou kontrolného súčtu. Kontrolný súčet sa počíta z celej správy a pri zmenení akejkoľvek hodnoty v správe sa zmení aj hodnota kontrolného súčtu. V súčasnosti existuje niekoľko algoritmov ako napríklad **SHA-1** alebo **MD5**, ktoré sa líšia samotným algoritmom a veľkosťou výstupu kontrolného súčtu.

Jednosmerná funkcia sa v kryptografii používa hlavne pri *elektronickom podpise*, ktorý zaisťuje nezmenený obsah správy a nemožnosť odmietnuť pôvod správy. Algoritmus pre elektronický podpis je nasledujúci: text správy sa pomocou jednosmernej funkcie zahashuje a výsledná hodnota sa zašifruje privátnym kľúčom. Zašifrovaný *hash* predstavuje elektronický podpis, pretože akákoľvek zmena v texte správy sa prejaví v odlišnosti tohto podpisu. Príjemca správy s elektronickým podpisom najprv správu zahashuje rovnakým algoritmom ako odosielateľ. Na prijatý elektronický podpis použije verejný kľúč a tým získa pôvodný hashovací reťazec. Obe tieto hashovacie hodnoty porovná a iba v prípade zhody je možné dôverovať obsahu správy a potvrdiť totožnosť odosielateľa [16]. Tento algoritmus bližšie ilustruje obrázok 4.2.

Kapitola 5

Návrh aplikácie

Návr aplikácie pre zabezpečený prenos dat na mobilných telefónoch typu smartfone je rozdelený na dva základné celky. Prvý celok predstavuje vytvorenie zabezpečovacích kľúčov, pomocou ktorých budú dve mobilné zariadenia bezpečne komunikovať. Tieto kľúče predstavujú základný stavebý prvok zabezpečenej komunikácie a bez nich nie je možné bezpečne komunikovať. Získanie týchto kľúčov sa uskutoční v niekoľkých krokoch:

1. **Vygenerovanie dvojice privátneho a verejného kľúča** - tento krok sa deje iba počas počiatočnej inicializácie aplikácie. Tento krok musia vykonať obe komunikujúce strany. Táto dvojica slúži na podpisovanie prenesených dát a na overenie integrity dát. V nasledujúcich krokoch sa pošle verejný kľúč druhej komunikujúcej strane. Privátny kľúč sa nesmie nikam odoslať.

2. **Vygenerovanie symetrického kľúča relácie** - tento kľúč slúži na šifrovanie väčšieho objemu dát medzi dvoma zariadeniami. Toto generovanie vykoná iba tzv. *Master zariadenie*, ktoré riadi komunikáciu a kľúč prepošle druhej komunikujúcej strane cez zabezpečený kanál.

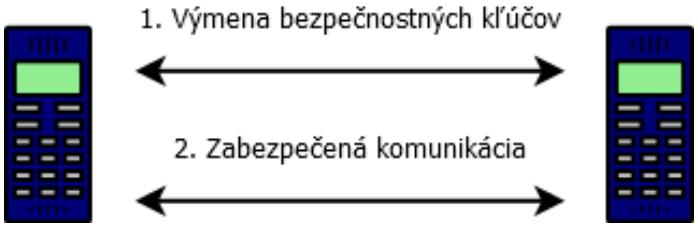
3. **Výmena kľúčov** - pomocou bezpečného prenosu sa pošle symetrický kľúč druhej komunikujúcej strane. Prenesie sa symetrický kľúč relácie a asymetrický verejný kľúč. Druhá komunikujúca strana taktiež odošle svoj verejný kľúč.

4. **Uloženie kľúčov do databázy** - každá komunikujúca strana si uloží do svojej databázy verejný kľúč a kľúč relácie tej druhej komunikujúcej strany. Pred komunikáciou a šifrovaním dát sa tieto kľúče získajú z databáze a sú použité pri patričných šifrovacích algoritmoch.

V druhom kroku sa bude na základe zabezpečovacieho kľúča komunikovať po jednotlivých komunikačných kanáloch. V závislosti na použitom komunikačnom kanále (Bluetooth, NFC, SMS, MMS atď.) sa zvolí miera zabezpečenia, keďže niektoré kanály sú limitované prenosovou rýchlosťou alebo objemom prenesených dát. Viac v kapitole **??**.

5.1 Rozdelenie aplikácie na moduly

Celá aplikácia je navrhnutá tak, aby oddeľovala užívateľské prostredie od zabezpečovacej logiky. Užívateľské prostredie disponuje GUI rozhraním, ktoré obsahuje prvky na nastavenie celej aplikácie a stará sa v prevažnej väčšine o získanie dát od užívateľa. Tieto dáta predáva



Obrázok 5.1: Zabezpečená komunikácia.

zabezpečovacej vrstve, ktorá zašifruje dáta a vráti šifrované dáta GUI prostrediu alebo ich ďalej odošle komunikačným kanálom. Zabezpečovacia vrstva zabaľuje komunikačné kanály ako je Bluetooth, NFC alebo GSM modul pre použitie s inými inteligentnými telefónmi. Obsahuje databázu kľúčov a funkčnosť s ňou spojenú, šifrovacie algoritmy a implementáciu protokolov na výmenu kľúčov a dát. Pre aplikácie je dostupné iba rozhranie umožňujúce prenos dát.

Návrh zabezpečovacej vrstvy sa skladá z niekoľkých základných modulov, ktoré implementujú celú funkčnosť. Prvou časťou je modul na správu kľúčov. V ňom sú uložené verejné a relačné kľúče všetkých kontaktov a verejný a súkromný kľúč aplikácie. Taktiež dovoľuje generovať nové kľúče alebo ich naopak odstraňovať. Na tento účel je vytvorených niekoľko metód.

Ďalším modul spravuje zariadenia ako je Bluetooth, NFC alebo GSM. Ten umožňuje pristupovať k zariadeniam a nadväzuje komunikáciu s inými zariadeniami. V prípade GSM modulu umožňuje taktiež odosielať a prijímať SMS správy, MMS správy. Taktiež posiela notifikáciu na vyššiu úroveň, aby aplikácia postavená nad zabezpečovacou vrstvou vedela o prichádzajúcej správe.

Nasledujúci modul slúži na šifrovanie a dešifrovanie správ. Implementuje rôzne šifrovacie, dešifrovacie a hashovacie algoritmy. Tento modul ma prístup k modulu pre správu kľúčov a dokáže si získať potrebné kľúče na základe rôznych parametrov ako sú napríklad telefónne číslo, meno kontaktu atď. Taktiež obsahuje potrebné algoritmy na automatické zašifrovanie a podpis správy.

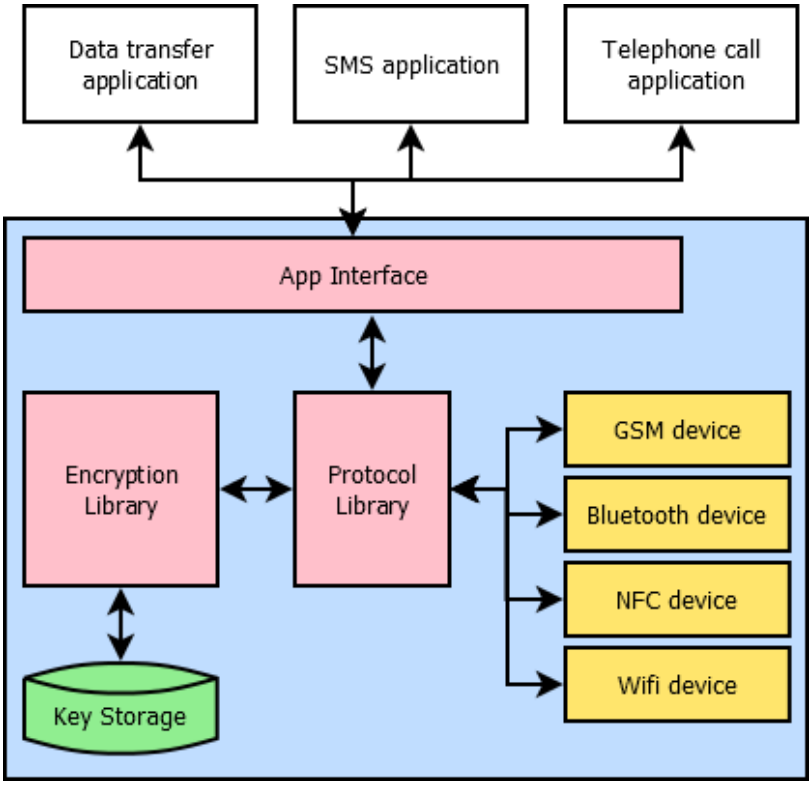
Pre odosielanie a prijímanie správ je potrebné implementovať modul, ktorý sa stará o protokolovú komunikáciu. V ňom sú implementované protokoly slúžiace na komunikáciu na vyššej úrovni. Vstupom do tohto modulu je zašifrovaná správa a podpis správy. Modul sa postará o poslanie zašifrovanej správy a podpisu v správnom poradí. Taktiež v opačnom prípade umožní príjem zašifrovanej správy a podpisu správy. Tento modul komunikuje priamo s komunikačnými kanálmi ako je Bluetooth, NFC alebo GSM.

Posledným modulom je rozhranie zabaľujúce celú funkčnosť zabezpečeného kanálu a umožňuje aplikáciám na najvyššej vrstve prístup k poslaniu správy a príjem správy, pričom celá zabezpečovacia funkčnosť je pre aplikáciu skrytá.

Popis jednotlivých modulov ilustruje obrázok 5.2

5.2 Vytvorenie bezpečnostného kľúča

Základným stavebným prvkom pri bezpečnej komunikácii je vytvorenie bezpečnostného kľúča, na základe ktorého budú komunikačné strany komunikovať. Na základe predošlého popisu komunikačných kanálov a na základe vytvorenia čo najbezpečnejšieho systému je najvhodnejšie stanoviť prioritu výmeny kľúča pomocou komunikačného kanálu. Priorita je



Obrázok 5.2: Návrh modulov aplikácie.

stanovená na základe čo najmenšej vzdialenosti komunikačných strán, aby sa eliminovala možnosť odpočutia prenesenej informácie pomocou rádiového signálu. Z bezpečnostného hľadiska je najvhodnejšie použiť na výmenu kľúčov technológiu NFC, ktorá je určená na bezpečnú komunikáciu. Bezpečnosť tejto technológie je vyššia ako u Bluetooth, pretože dosah signálu je len 20 cm, čo znemožňuje prípadnému útočníkovi zachytiť komunikáciu.

Je nutné brať na vedomie, že so znižujúcou sa prioritou narastajú bezpečnostné riziká zachytenia kľúča a možnosť zneužitia. Pokiaľ nie je technológia NFC podporovaná mobilným telefónom, tak je priorita komunikačného kanálu stanovená nasledovne:

1. **NFC** - najvyššia priorita pre výmenu kľúčov.
2. **Bluetooth** - stredná priorita pre výmenu kľúčov.
3. **GSM** - najnižšia priorita pre výmenu kľúčov.

Výmena kľúča by mohla byť uskutočnená aj pomocou internetu, ale v tomto prípade figuruje v komunikácii aj server, ktorý nemusí byť dostatočne zabezpečený. Z tohto dôvodu je najvhodnejšie použiť iba lokálnu komunikáciu.

Uloženie bezpečnostných kľúčov v aplikácii je riešené pomocou internej databáze. Prístup k tejto databáze je možný iba pomocou aplikácie, ktorá ju vytvorila. Ostatné aplikácie majú odoprené právo prístupu. Každý bezpečnostný kľúč je nutné asociovať s daným kontaktom telefónu pre lepšiu správu alebo notifikáciu užívateľa o exspirácii šifrovacích kľúčov.

V prípade, že ku kontaktu nie je priradený žiaden šifrovací kľúč, tak je o tom informovaná aplikácia a užívateľ. Na základe toho sa uskutočnia patričné kroky. Dáta sa pošlú

nezabezpečene alebo sa nepošlú vôbec. Tento prípad môže nastať iba v prípade tzv. *offline* komunikácie, pretože pri tzv. *online* komunikácii sa vždy vytvorí nový kľúč relácie. Bližšie je táto problematika popísaná v nasledujúcej podkapitole 5.2.1.

Android aplikácie môžu taktiež využiť tzv. *KeyStore* démona, ktorý dokáže ukladať dvojice kľúčov (verejný a súkromný), ale v našom prípade potrebujeme uložiť iba verejné kľúče čo nieje uvedeným démonom podporované.

5.2.1 Rýchlosť a objem dát prenosu

Jadro aplikácie bude pracovať na viacerých vrstvách bezpečnosti podľa toho na aký účel sa bude používať bezpečný prenos. Niektoré komunikačné kanály nedovoľujú posielať veľké objemy dát alebo neposkytujú dostatočnú prenosovú rýchlosť. Z toho dôvodu je vhodné rozdeliť zabezpečenie na niekoľko úrovní. Prvou úrovňou je plné zabezpečenie, pri ktorom sa využíva všetkých zabezpečovacích mechanizmov a to pomocou šifrovania dát a elektronického podpisu (online).

Druhou úrovňou je o niečo nižšie zabezpečenie, ktoré bude poskytovať iba šifrovanie dát bez elektronického podpisu (offline). Pred samotnou komunikáciou na tejto úrovni musí predchádzať výmena šifrovacích kľúčov, keďže komunikačný kanál je limitovaný. Výmena kľúčov sa môže diať aj za pomoci iných kanálov, ktoré boli popísané v kapitole o vytvorení zabezpečovacích kľúčov 5.2. V prípade, že by neboli pred komunikáciou vytvorené šifrovacie kľúče a účastník by vyžadoval šifrovanie, tak ho aplikácia explicitne požiada o výmenu kľúčov alebo upozorní užívateľa na skutočnosť, že dáta požadované k odoslaniu nebudú šifrované.

Toto rozdelenie na dve úrovne zabezpečenia bolo pomenované na **online** komunikáciu pre prvú úroveň a **offline** komunikáciu pre druhú úroveň. Tieto názvy vyjadrujú prístup k výmene šifrovacích kľúčov. Na online úrovni existuje spôsob, ktorým je možné vymeniť šifrovacie kľúče a to ten, že využijeme už existujúci kanál. Pri každej ďalšej online komunikácii sa vygenerujú vždy nové kľúče relácie, aby bola zaistená dostatočná obmena kľúčov.

Pri offline úrovni tento spôsob výmeny šifrovacích kľúčov nemáme k dispozícii, a preto aplikácia predpokladá aj online komunikáciu. V prípade, že by k online komunikácii nedochádzalo dlhší čas, tak by bol užívateľ vždy požiadaný o explicitnú online komunikáciu a bol by aplikáciou upozornený na vypršanie doby platnosti kľúča relácie. Šifrovanie by bolo zaistené aj po skončení doby platnosti kľúča len by dochádzalo k upozorneniu užívateľa.

5.3 Bezpečná komunikácia

Celý koncept zabezpečeného prenosu dát spočíva v tom, že sa vytvorí medzivrstva, ktorá zabalí komunikačné moduly poskytujúce komunikáciu s inými inteligentnými telefónmi a umožní šifrovací prenos medzi aplikáciami na týchto telefónoch. Táto medzivrstva obsahuje úložisko šifrovacích kľúčov a poskytuje programové rozhranie pre vyššie vrstvy, ktoré vyžadujú zabezpečenie. Takýmito aplikáciami môžu byť napríklad šifrované SMS správy, šifrovanie prenášaných súborov alebo šifrovanie pri prenose kontaktov.

Šifrovanie a dešifrovanie je zabezpečené pomocou niekoľkých algoritmov. Pre symetrickú kryptografiu je použitý algoritmus AES. Tento algoritmus je podporovaný u väčšiny inteligentných telefónov a vo veľkej miere sa používa aj u bezdrôtových sietí napr. u WAP2 technológie. Jedná sa o jeden z najnovších algoritmov, u ktorého boli odstránené nedostatky v porovnaní s inými algoritmami ako napríklad DES [19].

Pre asymetrické šifrovanie je použitý algoritmus RSA, ktorý dokáže šifrovať aj dešifrovať správy a môže byť použitý aj u elektronického podpisu správy. Tento algoritmus dokáže vhodne zabezpečiť správu s dostatočne dlhým kľúčom.

Pre zabezpečenie správy elektronickým podpisom je nutné použiť hašovaciu funkciu. Najlepším kandidátom je metóda SHA1, ktorá poskytuje dostatočnú rýchlosť aj bezpečnosť. Čo znemožňuje útočníkom narušiť integritu správy bez toho, aby to príjemca správy nezistil [15].

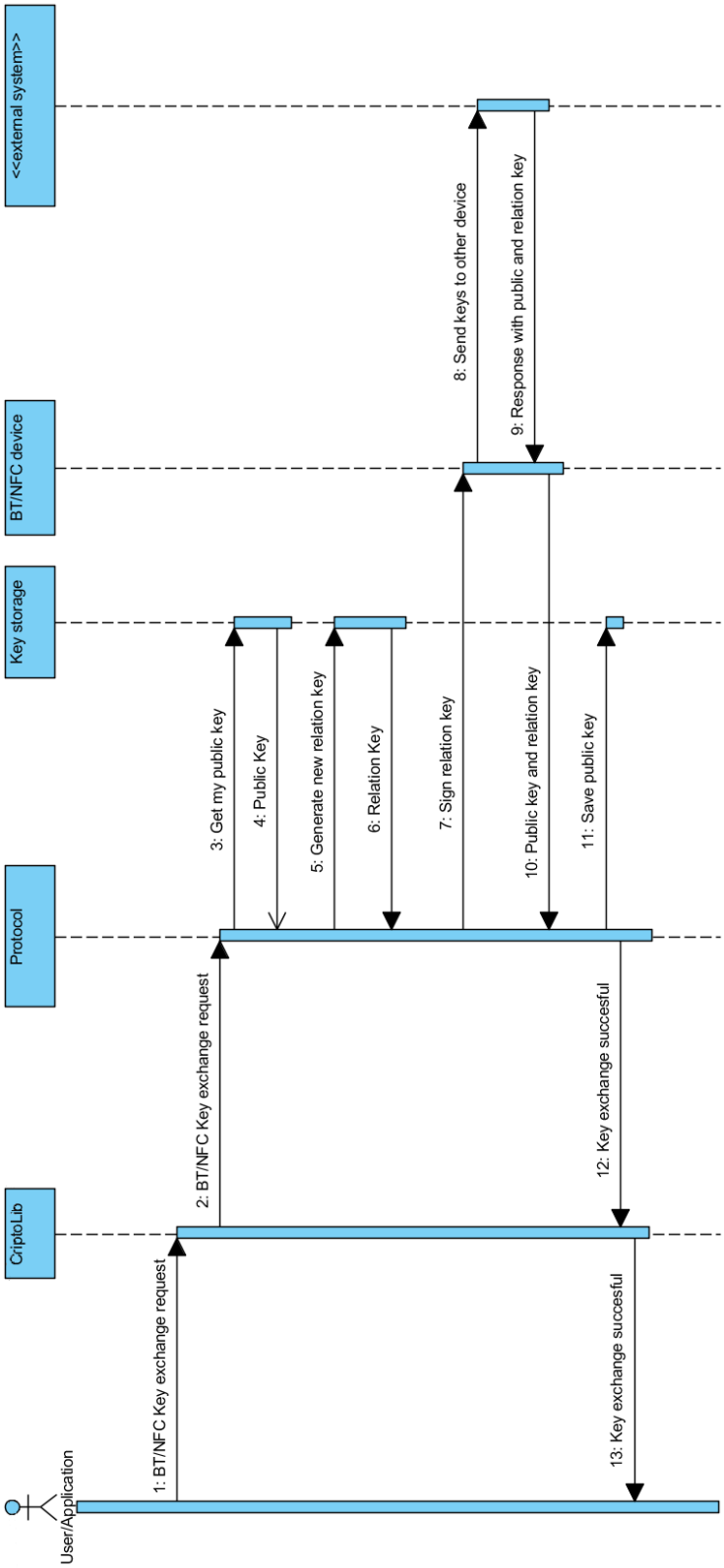
5.3.1 Návrh protokolu pre výmenu kľúčov

Aplikácia na šifrovanie dát nedokáže správne fungovať bez šifrovacích kľúčov. Na šifrovanie dát sú použité dva druhy kľúčov uložené v lokálnej databáze. Aby mohli komunikujúce strany bezpečne komunikovať je nutné vymeniť verejne šifrovacie kľúče a jedná komunikujúca strana musí vytvoriť kľúč relácie, s ktorým sa budú šifrovať väčšie objemy dát.

Poslanie dát pomocou SMS správy by bolo asi najviac užívateľsky prívetivé, keďže by stačilo odoslať kľúče druhej komunikujúcej strane a ona by odpovedala svojím verejným kľúčom. Po tejto výmene by mohli komunikujúce strany bezpečne komunikovať, lenže táto implementácia naráža na niekoľko problémov. Jedným z nich je, že pri SMS správe je pomerne jednoduché podvrhnúť identitu odosielateľa. Na tento účel existuje niekoľko verejne dostupných serverov, ktoré dokážu „predstierať“ inú identitu odosielateľa. Príjemca takejto správy nedokáže s istotou povedať, či sa jedná o daného užívateľa [10].

Druhou nevýhodou je, že posielanie šifrovacích kľúčov pomocou SMS nie je bezpečné keďže sú dáta posielane v otvorenom tvare cez verejnú sieť. Na zabezpečenie takejto siete by bolo treba posielanie niekoľkých správ čo by nebolo veľmi efektívne z finančného pohľadu. Ako bolo spomenuté v úvode tejto kapitoly, najlepším spôsobom výmeny kľúčov je pomocou technológii Bluetooth a NFC. Zjednodušený protokol, ktorým sa uskutočňuje výmena šifrovacích kľúčov znázorňuje nasledujúci obrázok 5.3.

Na obrázku sa nachádza niekoľko komponent, do ktorých je rozdelená zabezpečovacia vrstva. Výmenu šifrovacích kľúčov iniciuje užívateľ alebo aplikácia, ktoré vydá príkaz na výmenu kľúčov. Tento príkaz získa trieda implementujúca rozhranie, ktorá prepošle požiadavku do triedy implementujúcej posielanie dát na zariadenia. Táto trieda sa postará o získanie kľúčov z databázy alebo zaistí aby sa vygenerovali nové kľúče a tieto kľúče odošle do daného zariadenia. Výber zariadenia musí byť špecifikovaný pred samotným poslaním dát. Zariadenie sa postará o prenos kľúčov druhej komunikujúcej strane. Tá uloží obdŕžané kľúče do svojej databáze. Druhá komunikujúca strana odošle svoj verejný šifrovací kľúč strane, ktorá iniciovala komunikáciu. Tá uloží verejný kľúč vo svojej databáze. Zabezpečovacia vrstva po uložení kľúčov notifikuje užívateľa alebo aplikáciu, ktorá využíva šifrovaciu vrstvu, o úspešnosti výmeny kľúčov.



Obrázok 5.3: Návrh protokolu na výmenu šifrovacích kľúčov.

5.4 Návrh aplikácie využívajúcej zabezpečovaciu vrstvu

Neoddeliteľnou časťou celej aplikácie je časť, ktorá sa stará o komunikáciu s užívateľom a využíva mechanizmov popísaných v predošlých podkapitolách. Jej úlohou je informovať užívateľa o prichádzajúcich udalostiach a v prípade náznakoch porušenia bezpečnosti notifikovať upozornením. Porušenie bezpečnosti môže nastať pri exspirácii šifrovacích kľúčov alebo v prípade snahy o podvrhnutie nesprávnych šifrovacích kľúčov. Všetky tieto udalosti musia viesť k tomu, že sa o tom dozvie užívateľ a vykoná patričné kroky vedúce k vyriešeniu problémov.

Ďalšou úlohou užívateľského rozhrania je možnosť zadávania dát určených k šifrovaniu. Tieto dáta môžu byť použité na priamu komunikáciu s ďalšími subjektmi vo forme správy alebo vo forme toku dát. Pokiaľ aplikácia vyžaduje uloženie šifrovaných dát, ktoré zadal užívateľ, tak je potrebné takúto funkcionalitu vytvoriť, pretože zabezpečovacia vrstva neponúka túto možnosť. Najvhodnejší spôsob je použiť *content provider*, pracujúci s *aktivitami*. Táto kombinácia zaručí, že pri zmene dát v databáze sa aktualizuje aj obsah aktivity. Alternatívou k tomuto spôsobu by mohlo byť vytvorenie samotnej SQLite databáze a pri každej zmene v databáze explicitne vykresliť obsah aktivity. Tento spôsob je pomerne zložitý na ošetrenie všetkých stavov v životnom cykle aktivity. Užívateľské rozhranie by malo implementovať možnosť nastavenia prostredia a celej aplikácie. Zabezpečovacia vrstva obsahuje niekoľko nastavení, ktoré je nutné ošetriť pri používaní v reálnych aplikáciách a na tento účel je nutné vytvoriť triedu, ktorá sa bude starať o tento prípad.

Kapitola 6

Implementácia aplikácie

Na základe získaných poznatkov z predošlých kapitol som sa rozhodol implementovať aplikáciu, ktorá bude šifrovať SMS správy cez GSM sieť. Dôvody, ktoré ma k tomu viedli sú, že GSM sieť neposkytuje šifrovanie end-to-end a prenos SMS správ je pomerne rozšírený.

Aplikácia je implementovaná ako Android aplikácia do dvoch balíčkoch. Prvý balíček predstavuje samotnú SMS aplikáciu, ktorá poskytuje prevažne užívateľské rozhranie, triedy na odosielanie a prijímanie správ a volá funkcionalitu druhého java balíčku.

Druhý balíček slúži na šifrovanie a dešifrovanie predložených dát, taktiež obsahuje implementáciu rôznych protokolov na ustanovenie kľúčov a prenos dát pomocou Bluetooth alebo NFC, implementuje úložisko bezpečnostných kľúčov a poskytuje programové rozhranie vyššej vrstve, ktorá ho využíva. Táto vrstva zabaľuje bezpečnostné mechanizmy pred vyššou vrstvou.

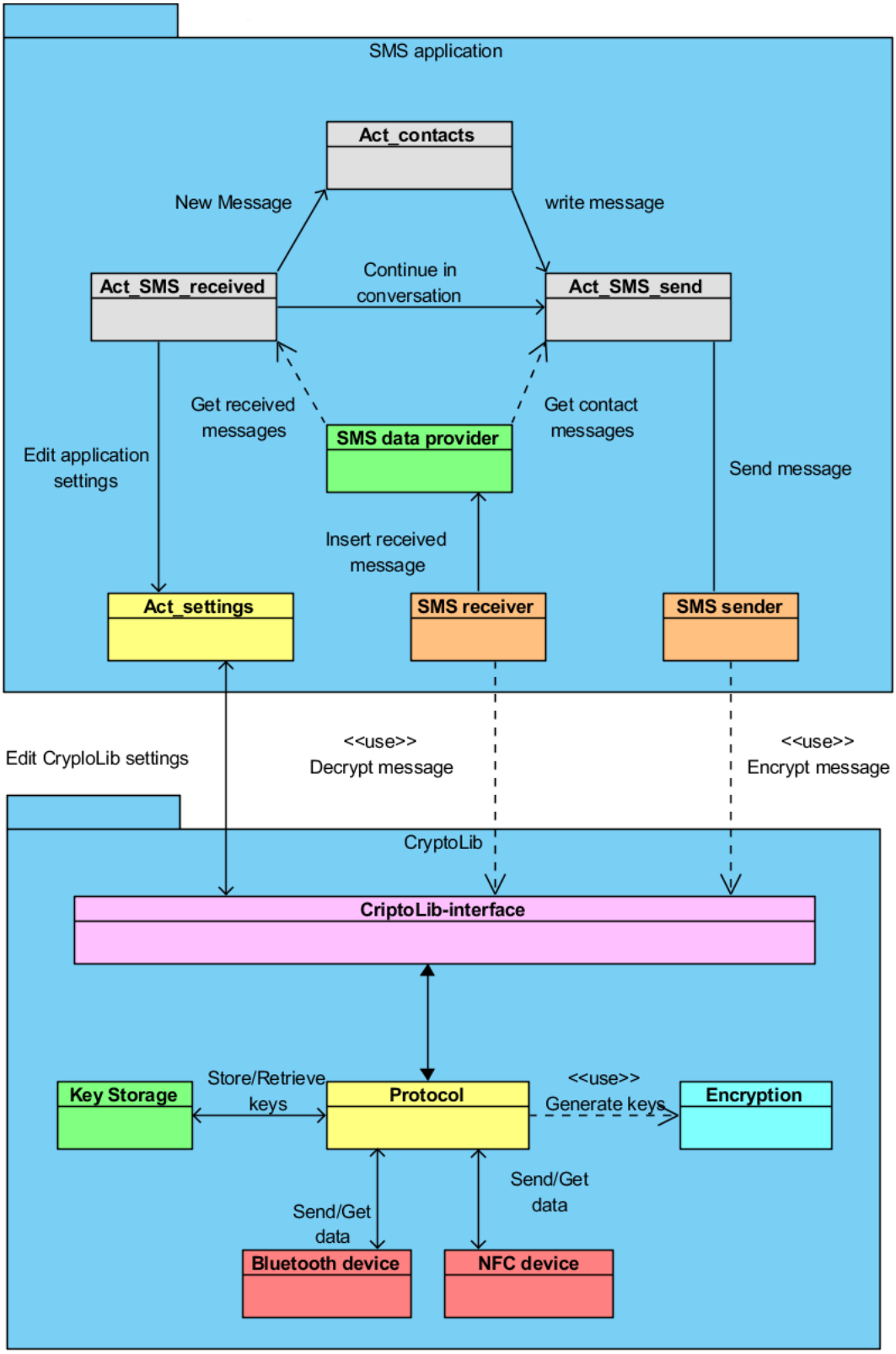
6.1 Návrh tried zabezpečenej komunikácie

Implementácia zabezpečenej komunikácie je rozdelená do tried, ktoré dekomponujú funkcionalitu a umožňujú jednoduchšiu správu aplikácie. Trieda poskytujúca rozhranie pre SMS aplikáciu ma názov `SecuredCommunication`. Trieda obsahuje metódy pre nastavenie zariadenia, cez ktoré sa bude uskutočňovať výmena kľúčov, nastavenie kontaktu, s ktorým sa bude komunikovať, poslanie dát a príjem dát z/do zariadenia. Táto trieda odtieňuje zabezpečenie od užívateľa, prípadne vývojára aplikácie, ktorý používa zabezpečenú komunikáciu. Trieda umožňuje prístup k samotným zabezpečeným komunikačným kanálom.

Základnou triedou pre šifrovanie a dešifrovanie správ tvorí trieda `CryptoLib`, ktorá implementuje všetky potrebné algoritmy. Táto trieda umožňuje aj elektronický podpis správy na základe súkromného kľúča aplikácie. Všetky kľúče sú získavané z triedy implementujúcej databázu kľúčov `DataProvider`. Trieda CryptoLib taktiež dokáže vytvárať nové kľúče relácii alebo dvojicu súkromný a verejný kľúč, v prípade exspirácie kľúča relácie .

Trieda `DataProvider` sa stará o poskytovanie verejných kľúčov uložených v databáze. Využíva pri tom SQLite databázu, ktorá sa nachádza v operačnom systéme. Táto trieda dokáže na základe kontaktu získať verejný kľúč a kľúč relácie, uložiť ich v databaze a poskytnúť ich triede `CryptoLib`. Táto trieda je implementovaná ako Content Provider s explicitne zakázaným poskytovaním dát iným aplikáciám.

Srdcom zabezpečenej vrstvy je trieda Protocol, ktorá implementuje posielanie dát v správnom formáte na dané zariadenie. Do tejto triedy vstupujú už zašifrované dáta z triedy `SecuredCommunication` a taktiež v prípade príjmu dát zo zariadenia sa preposielajú do



Obrázok 6.1: Návrh tried aplikácie a zabezpečovacej vrstvy.

tejto triedy. Implementuje rôzne protokoly na výmenu kľúčov a posielanie dát na kanály.

6.2 Užívateľské rozhranie SMS aplikácie

Aplikácia demonštrujúca využitie zabezpečenej vrstvy sa skladá z niekoľkých tried, ktoré sa starajú o užívateľské rozhranie, posielanie SMS správ. Aplikácia obsahuje triedu `DataProvider`, ktorá obsahuje všetky prijaté a odoslané správy a je implementovaná ako Content Provider.

Užívateľské rozhranie je tvorené triedami `Activity_msg_recv`, `Activity_contacts`, `Activity_options` a `Activity_msg_send`, ktoré sú navzájom prepojené a bežný užívateľ má možnosť k ním prístupovať. Obsahujú GUI prvky umožňujúce jednoduchšiu manipuláciu s prijatými správami a konfiguráciu aplikácie. Aktivity predstavujú jediné rozhranie, ktorým môže bežný užívateľ pristupovať k aplikácii.

Najdôležitejšia trieda je `SMS_bcst_receiver`, ktorá slúži na príjem správy od operačného systému. Aplikácia na základe tejto správy dozvie o prijatí správy a zavolá patričné metódy na dešifrovanie správy. Správa je ďalej zapísaná do `DataProvider` triedy, ktorá ju uloží a vyvolá obnovenie aktivít na prekreslenie svojho obsahu.

Aktivita `Activity_contacts` slúži na vyhľadávanie kontaktov a na indikáciu či je možné s daným kontaktom šifrovane komunikovať. Taktiež poskytuje informáciu o exspirácii relačného kľúča a ďalej umožňuje vymeniť bezpečnostné kľúče s iným mobilným telefónom.

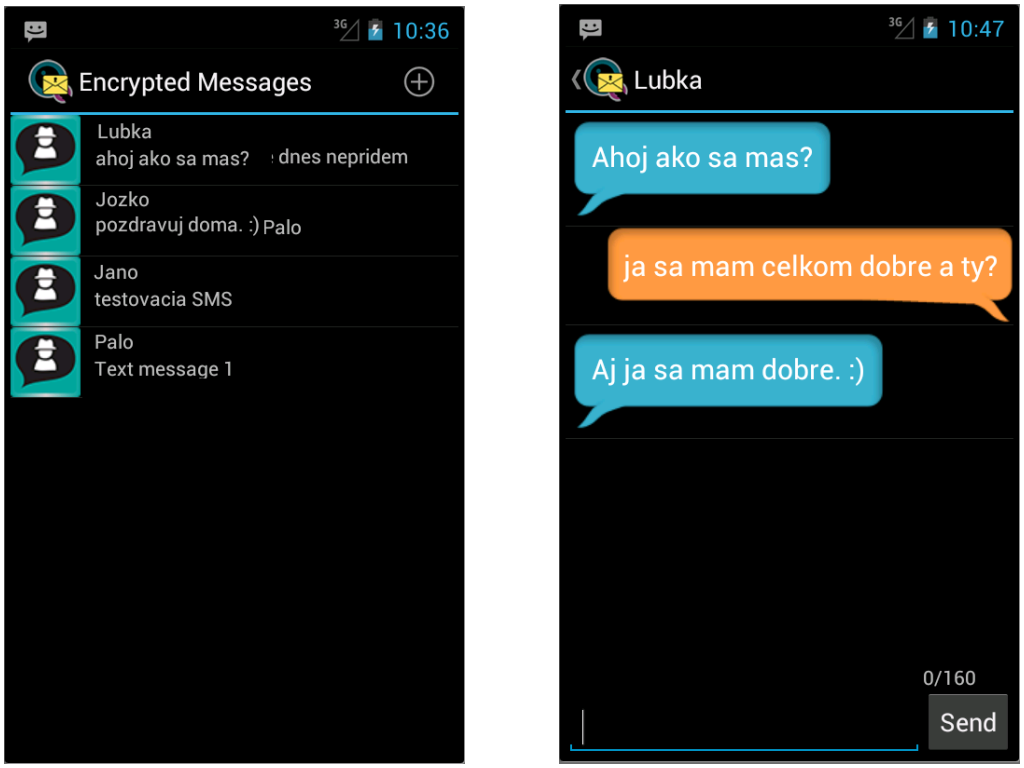
Aktivita `Activity_options` umožňuje konfiguráciu aplikácie a prípadne konfiguráciu zabezpečeného kanála.

Kapitola 7

Testovanie implementácie

Aplikácia bola vyvíjaná vo vývojovom prostredí Eclipse za použitia emulátoru operačného systému Android vo verzii 4.2. Emulátor dokáže emulovať rôzne verzie operačného systému Android. Z jeho pomocou bolo možné overiť správnu funkčnosť mnohých komponent ako je napríklad databázový model, šifrovací systém, generovanie šifrovacích kľúčov, posielanie správ do druhého emulátoru a taktiež otestovať správne zobrazenie jednotlivých aktivít.

V emulátore je možné dokonca simulovať posielanie SMS správ do druhého emulátoru. Toto je možné pomocou zadania portu, na ktorom operuje daný emulátor, namiesto čísla odosielateľa. Tak bolo možné overiť správnu funkčnosť posielania zašifrovaných SMS správ bez plytvania finančných prostriedkov.



Obrázok 7.1: Návrh aktivít pre SMS aplikáciu.

Jedinou nevýhodou emulátoru je v tom, že nedokáže simulovať komunikáciu pomocou

zariadení ako je Bluetooth a NFC. Na otestovanie správnej funkčnosti je nutné aplikáciu nainštalovať na reálne zariadenia. Testovanie Bluetooth zariadenia bolo uskutočnené na zariadení *Sony Xperia Sola* a *Sony Xperia Miro*, pričom sa overila funkčnosť výmeny šifrovacích kľúčov a posielanie šifrovaných SMS správ. Overenie výmeny šifrovacích kľúčov pomocou zariadenia NFC sa nepodarilo z dôvodu nedostatku zariadení s touto technológiou. Aj napriek tomu bolo možné otestovať základnú funkcionalitu na mobilnom telefóne *Sony Xperia Sola*, ktorý disponuje NFC modulom.

Kapitola 8

Záver

Cieľom tejto práce bolo preskúmať možnosti mobilných telefónov typu smartphone a technológií, ktorými je možné zabezpečiť šifrovanú komunikáciu. V súčasnej dobe je najrozšírenejší operačný systém Android, pod ktorým som sa rozhodol bližšie rozobrať danú problematiku a preskúmať jeho programové možnosti. Na základe získaných poznatkov o mobilných telefónoch a komunikačných kanáloch bol vypracovaný návrh zabezpečenej komunikácie. Tento návrh je možné aplikovať aj pre iné mobilné operačné systémy. Na demonštráciu šifrovanej komunikácie bola navrhnutá a následne implementovaná aplikácia, ktorá umožní odosielať a prijímať šifrované SMS správy.

Aplikácia na šifrovanie SMS správ potrebuje k správnemu chodu šifrovacie kľúče druhých komunikujúcich strán. K tomuto účelu bolo využitých niekoľko technológii ako napríklad Bluetooth alebo NFC pre ich výmenu. Tento spôsob bol zvolený z dôvodov eliminácie možného odpočutia rádiového signálu a útoku „man in the middle“. Na výmenu šifrovacích kľúčov by sa dala využiť GSM sieť napríklad v podobe SMS správy, ale v tomto prípade sa jedná o nebezpečný spôsob pretože s dátami môže disponovať mobilný operátor.

Testovanie aplikácie ukázalo, že SMS aplikáciu so zabezpečovacou vrstvou je možné nasadiť do reálneho prostredia a umožniť užívateľom šifrovanú komunikáciu medzi mobilnými telefónmi typu smartphone. Táto aplikácia umožňuje posielať šifrované SMS správy a zabezpečiť šifrovanie nie len v určitej časti siete ale počas celého prenosu až ku koncovému užívateľovi tzv. end-to-end zabezpečenie. Užívateľ ma na výber z niekoľkých možností ako si vymení s druhou komunikujúcou stranou šifrovacie kľúče. Tým umožňuje šifrovane komunikovať aj mobilným telefónom, ktoré nie sú na vrchole technologického reťazca a zastávajú nižšiu cenovú vrstvu mobilných telefónov.

Na druhej strane technológia prenosu dát medzi mobilnými telefónmi nie je ešte dostatočne vyvinutá, keďže nedisponuje možnosťou posielania väčšieho množstva dát na iný mobilný telefón alebo nedisponuje technológiou, ktorá by dokázala vytvoriť priamy kanál na posielanie dát medzi mobilnými telefónmi. Priamy prenos údajov by bol možný iba cez server, ktorý by dokázal posielať požiadavky na spojenie pre mobilné telefóny. V tomto prípade by museli byť mobilné telefóny konštantne pripojene k tomuto serveru a „vyzdvihovať“ si požiadavky od tohto serveru aby bolo možné nastoliť určitý druh komunikácie. Tento prístup je ale značne neefektívny a nie je priamy ako napríklad u technológie Bluetooth.

Samotná zabezpečovacia vrstva bola navrhnutá tak, aby mohla byť použitá s inými mobilnými aplikáciami. Obsahuje databázu šifrovacích kľúčov pre jednotlivé kontakty, čím by sa mohla využiť aj u aplikácií sociálneho charakteru. Po menšej úprave by sa mohla vytvoriť aplikácia, ktorá by na základe šifrovacích kľúčov dokázala šifrovať osobné informácie na serveroch ako napríklad Facebook, Google, atď. a umožniť užívateľom súkromie.

Na týchto serveroch by boli všetky informácie uložené ako šifrované a jediným užívateľom, ktorý by mohol prečítať osobné informácie by boli kontakty, s ktorým si užívateľ vymení šifrovacie kľúče. Tým by sa stal mobilný telefón kľúčom ku prístupu k osobným informáciám uloženým na verejných serveroch.

Literatura

[1] Processing Standards Publication 197, Advanced Encryption Standard (AES). 2001.
URL http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[2] Appcelerator, Inc.: Appcelerator Platform.
URL http://www.appcelerator.com/

[3] Arndt, R. Z.: You Should Put Antivirus Software on Your Phone. 2012.
URL
http://www.popularmechanics.com/technology/how-to/computer-security/you-should-put-antivirus-software-on-your-phone-14886208

[4] Blackberry, Inc.: Runtime for Android apps.
URL http://developer.blackberry.com/android/tools/

[5] Enrico Angelini: 5 Pros and Cons of Appcelerator's Titanium.
URL http://enricoangelini.com/2012/5-pros-and-cons-of-appcelerators-titanium/

[6] Etherington, D.: Cool iPhone App: Start Your Car From Virtually Anywhere. 2000.
URL http://gigaom.com/apple/cool-iphone-app-start-your-car-from-virtually-anywhere/

[7] Google, I.: A Java to iOS Objective-C translation tool and runtime: j2objc. 2012.
URL http://gigaom.com/apple/cool-iphone-app-start-your-car-from-virtually-anywhere/

[8] Google, Inc.: Android reference manual.
URL http://developer.android.com/reference/packages.html

[9] Guillermo Matas Miraz, M. Á. G.-N., Irene Luque Ruiz: University of Things: Applications of Near Field. 2009.
URL http://www.inderscience.com/www/info/ijwi/art/tjew3101.pdf

[10] Harsh Agrawal: Send Anonymous and Fake sms.
URL http://www.shoutmeloud.com/send-anonymous-fake-sms.html

[11] Kardach, J.: Bluetooth Architecture Overview. 2013.
URL http://download.intel.com/technology/itj/q22000/pdf/art_1.pdf

[12] nearfieldcommunication.org: Security Concerns with NFC Technology.
URL http://www.nearfieldcommunication.org/nfc-security.html

[13] Nokia: Introduction to NFC. 2011.
URL http://www.developer.nokia.com/info/sw.nokia.com/id/bdaa4a0f-fcf3-4a4b-b800-c664387d6894/Introduction_to_NFC.html

[14] Perrin, C.: Encrypt calls on your Android device with RedPhone. 2011.
URL http://www.techrepublic.com/blog/security/encrypt-calls-on-your-android-device-with-redphone/5300

[15] Projects, W.: Standing Document 1: WG8 Projects.
URL http://wg8.de/sd1.html

[16] Pužmanová, R.: *Bezpečnost bezdrátové komunikace*. CP Books, a.s., 2005, iSBN 80-251-0791-4.

[17] Schiller, J.: *Mobile Communications*. Edinburgh Gate, Harlow: Pearson Education Limited, 2003, iSBN 0-321-12381-6.

[18] Statista: Market share held by smartphone operating systems worldwide in 2012 and 2016. 2012.
URL http://www.statista.com/statistics/182363/marketshare-forecast-of-smartphone-operating-systems/

[19] Wikipedia: Advanced Encryption Standard. 2013.
URL http://en.wikipedia.org/wiki/Advanced_Encryption_Standard

[20] Wikipedia: Kasumi. 2013.
URL http://en.wikipedia.org/wiki/KASUMI

[21] Zigurd Mednieks, G. B. M., Laird Dornin: *Programming Android*. O'Reily Media, 2011, iSBN 978-1-449-38969-7.

Príloha A

Obsah CD

CD Obsahujúce:

- Zdrojové súbory aplikácie.
- Elektronickú verziu tohto dokumentu.
- Spustiteľnú verziu aplikácie.
- Plagát

Príloha B

Plagát

ENCRYPTED MESSAGES

FACULTY OF INFORMATION TECHNOLOGY

Author: Dávid Bocko

Master's thesis
Supervised by Pavol Korček

Every user has a right for privacy !!

- **This application allows you to secure the SMS messages by end-to-end encryption.**
- **Using the AES and RSA encryption**
- **Supporting the Bluetooth and NFC key exchange**
- **User friendly GUI interface and much more...**